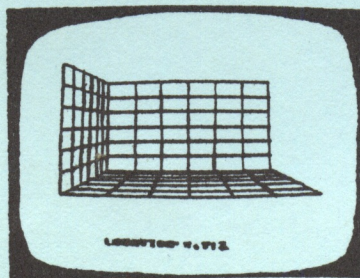*Sebree's Computing*

PRESENTS

3ᵈ DIMENSIONAL

COMPUTER GRAPHICS

ATARI 800

BASIC LANGUAGE

SEBREE'S COMPUTING

PRESENTS
3-DIMENSIONAL
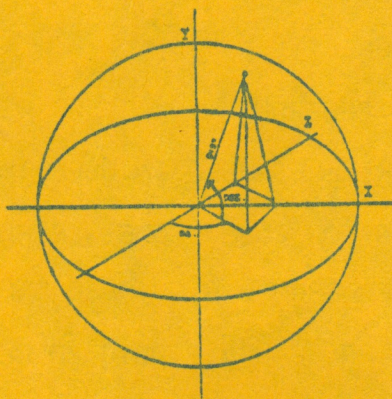# COMPUTER GRAPHICS

****ATARI 400/800****

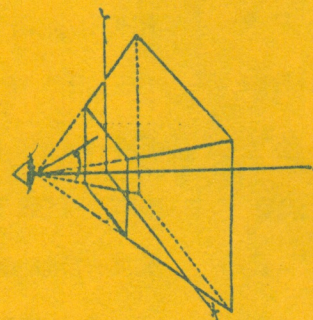(C)'80 Sebree's Computing-ATARI 800

BASIC VERSION
BY:
TIMOTHY HAYS

Registration number:

A-1227

© SEBREE'S COMPUTING
March 1980

456 Granite Ave.
Monrovia, CA 91016
213-359-8092

I M P O R T A N T   N O T I C E ! !

If, for some reason, your cassette program doesn't load, try advancing the tape a little (1 to 5 on the tape counter). This should remedy the problem in most cases. If you still have loading problems, it is because your ATARI 410 program recorder does not have the tape heads aligned properly. To align the tape heads, use the following procedure:

1) Insert the ATARI EDUCATION SYSTEM master cartridge.

2) Insert any ATARI EDUCATION SYSTEM cassette (Psychology, Physics, etc.) into your ATARI 410 program recorder.

3) Find a small screwdriver, (a jewelers screwdriver, etc.) which will fit in the tiny hole just above the playback head, (above the letter 'T' in the ATARI 410 logo). Some units may differ and/or the head-adjust screw may not be easily accessable. The screw is bracing the playback head, press down 'PLAY' and you may see it.

4) Press 'PLAY' on the recorder and press 'START' on the ATARI.

5) When the EDUCATION LESSON starts, watch for glitches on the TV screen. If there are glitches, adjust the screw with the screwdriver until the glitches are just about gone. Then, rotate the screw a little bit farther until you hear the DATA track (what you hear when you load in a program).

6) Now, listen very carefully and adjust the screw until the DATA track is just barely audible, this is the point which the two tracks (A & B) are separated so the audio track (the voice you hear) does not interfear with the DATA track.

Now, try loading that hard-to-load cassette. If it still doesn't load after doing this, try CLEANING and DE-MAGNETIZING the tape heads.

If the cassette still doesn't load after following ALL of the above listed procedures, then the tape itself is bad (dropout or glitch in tape, etc.), and you can return it for a new copy.

NOTE: SEBREE'S COMPUTING uses the TDK 'D' series 'Music Machine' high quality cassettes, SCOTCH brand cassettes, and BASF computer tape exclusevly.

NOTE note: It is advised that the user make himself a backup copy of the software so the user may not have any future problems, but please conform with all of the COPYRIGHT rules, that this is the only time the user may copy software— for his/her OWN USE. Copying software and selling it or giving it to a friend is illegal, against all COPYRIGHT rules, and may end in prosecution.

3-DIMENSIONAL COMPUTER GRAPHICS

CONTENTS

2 END

_ Since it's inception more than a decade ago, the field of computer graphics has captured the imagination, and technical interest of rapidly increasing numbers of individuals from all society. A high percentage of the growing ranks of computer graphics professionals has given primary attention to: computer-oriented problems in programming, system design, hardware, (not to mention Software), etc, etc,

_ A NEW and rapidly expanding field called 'Computer Graphics' is emerging.

_ Very FEW individuals know enough about the hardware, software, micro-processors, theory of design, the mathematics of computer graphics, color control, and the individual machine they are working with; in order to write, test, & debug 3-d graphics software effectively. This is why we, at SEBREE'S COMPUTING have decided to put MANY months work into this advanced graphics package. This package has a very powerful set of programs in which the applications start with major areas, such as: Management information, Scientific graphics, Command and control, Image processing, Real-time image generation & simulation, Electrical & mechanical design, and the applications just go on, and on.

_ The problem is: Each application area calls for markedly different hardware and software. So unique are those requirements that few existing systems can meet most of the demands posed by: even just a single class of applications. This program package HAS taken all of this into consideration: The software was written to be as versatile as possible, while still limiting itself in memory usage.

_ Most 3-D graphics users are interested primarily in using the software in their own special application. The design of the software made it fairly simple for the user to alter it, or use it as-is for his application. There are very few things that the user is required to know to use this package. You will have to recall some simple 3-D Coordinate Geometry that you learned in High School.

_ The ATARI uses the top-left corner of the screen as 0,0, but this software uses the very center of the screen for: 0,0,0 (X,Y,Z). The location YOU will be viewing the scene from will usually be 0,0,0. A data base (scene) in memory will usually be located at a depth (Z) of 100 or more, so the scene is in front of you. If the depth is equal to the viewers location (0), then you will end up standing inside of the object you wish to view! If this is the case, simply enter a negative depth for the viewers location, thus:
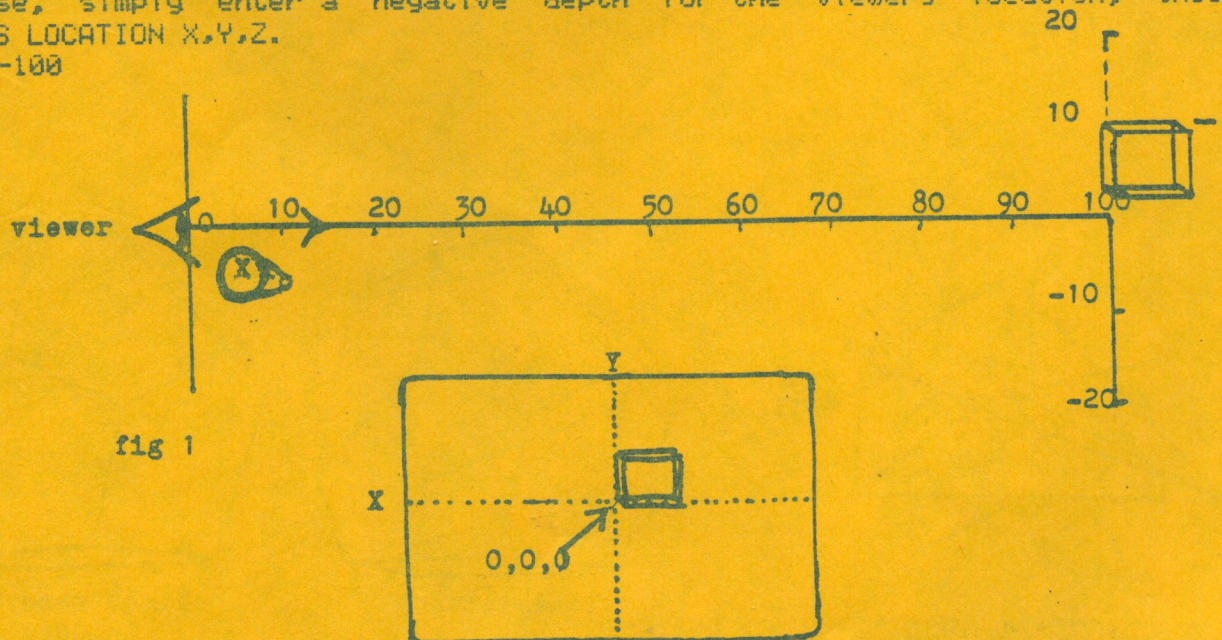VIEWERS LOCATION X,Y,Z.
?0,0,-100

fig 1

LINE CLIPPING AND PUSHING

_ One of the most important items in a Graphics package: the ability to draw objects that lie partially on the screen. This software will do just that with perfect precision, up to the last pixel on the very edge of the screen. When a line is partially on the screen, the program GOSUBs into a series of 'Clipping & Pushing' functions, until both edges of the line intersect the edges of the screen. This is the computation which takes the longest to perform, deciding which way to push the line, and how far.
_ Polygon clipping reduces a polygonal surface extending beyond the boundary of the screen. The process 'clips off' those parts of the polygon which lie outside the window. The perspective computations alone are inadiquate to meet most needs because they will erroneously project points behind the observer upside down and backwards onto the screen in front of him. The final division process in the perspective projection destroys one bit of signal information, and this is precisely the bit that keeps objects in front of an observer in front of him. If a line connects a point in front of an observer with a point behind him, it will project a very strange looking line if it's ends are simply projected by the ordinary rules.
Moreover, a point 'alongside' the observer, with the same value 'Z' as his eye, projects into a point with infinite values of 'X, Y, & Z', clearly a difficult point to represent. The point being, we must limit the values of 'X, Y, Z, & W' PRIOR to doing the division. If we are careful to limit the values of 'X, Y, Z, & W' appropriately, we can make the range of possible values for X,Y,& Z come out to be very simple. For example, it would be nice to have x & y in the range −1 to +1, because they would then be easy to scale to any avaliable precision in a subsequent depth computation.
This suggests that we always clip on the limits: $-w <= x <= +w$, $-w <= y <= +w$, $0 <= z <= +w$. These clipping limits correspond to six planes which we will call 'left', 'right', 'bottom', 'top', 'hither', and 'yon'. The six clipping planes form a truncated pyramid. In raw form, the transformation and clipping result in a fixed field of view. _ This package draws 'wire frame' objects only, as shown below. There is no hidden-line capability, simply because it takes too much time and memory to do so.

LINE CLIPPING & PUSHING

fig 2

Polyhedron...

lines intact

lines removed
(hidden)
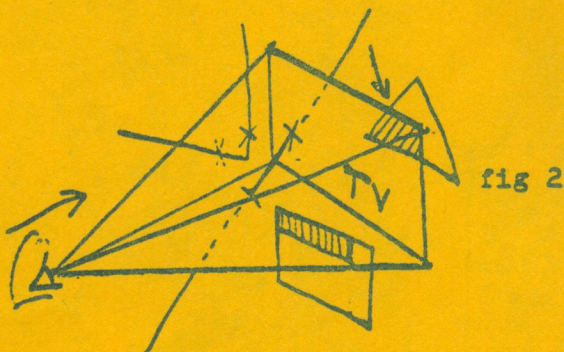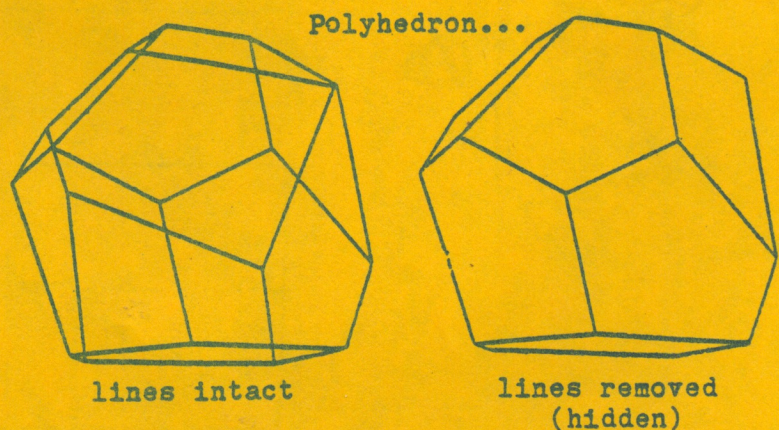
PERSPECTIVE PROJECTIONS

We are comcerned with the computer processing of three-dimensional data, to produce realistic two- dimensional pictures. When producing perspective pictures of opaque objects, we use depth information to perceive: which of the two objects is nearer the observer. We might do these computations before making the perspective projection, using the true depth of objects for comparison but we would then be troubled by the difficulty of computing: the reletive lateral position of objects considering the parallax of the viewpoint. It is generally much more convenient to make a three-dimensional perspective projection first, since this projection puts all of the objects into their correct positions on the screen,while preserving not only depth ordering but also the fact that straight lines project into straight lines.

In effect, the application of a proper perspective projection to a set of three- dimensional objects, enables us to treat the objects as if they were being viewed from infinity. The problem of available depth resolution has been a plagueing factor for some time. With hither and yon clipping, the available range of perspective depth is distributed as uniformly as possible, over the space between the hither and yon clipping planes. As you might expect, the closer the hither and yon clipping planes are to each other, the more depth resolution will be available in the space between them.

LOAD & GO.......

You're probably pretty anxious to start by now,so put in the tape or disk and load-in the first program..When the screen prompts you with 'READY', type RUN and press RETURN twice. This program is an intro with the titles of all the cassette programs in succession, there are six. This program uses user-defined text modes which will be explained later. This program will automatically load in the next program, the next should be loading in if you pressed RETURN twice.

PROGRAM>-- OVERSCAN.

Now type in RUN again and read the screen. This program is a neat little polygon plotter which we shall use to determine our needed 'overscan'. The OVERSCAN variable (OVS) is used to correct the problem we have with the pixels on the screen: NOT being perfectly square. If we put a square onto the screen measuring 40 X 40 pixels, it would really appear to be a rectangle. The overscan variable is used in conjunction with the software to aleviate this problem.

This is how you will use the program:

NUMBER OF SIDES?
?30
RADIUS OF PLOT? (HI-RES MAX=154)
?40
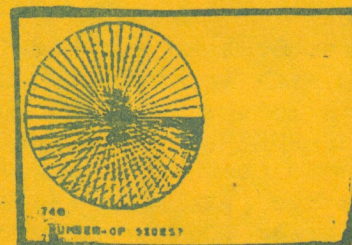DIAMETER PLOT ONLY? 'Y'=YES
?Y
GRAPHICS RESOLUTION? (MODE 3-8)
?7
CLEAR THE SCREEN? 'Y'=YES
?Y
COLOR? ALWAYS '1' FOR HIRES
?1

SAMPLE OUTPUT OF OVERSCAN PROGRAM

5

The program will now plot out a circle in graphics mode 7, with a color of
'1' (gold). If you have 8K memory, you will have to use graphics modes '5' or
'7' from now on (when the screen asks: SCREEN RESOLUTION?). If you have 16K
memory or more, RUN the above and replace GRAPHICS RESOLUTION with an '8'.
REMEMBER: Disk drive systems will have to add an extra 8K of memory to all of
these noted requirements!

_   If the circle is definetly not the same width as it is heighth, follow
the steps in REM lines 100-on, to re-adjust the overscan (OVS) so that the
circle is always symmetrical. Now, write this variable value down in this
manual so you will know what number to replace OVS with later. I suggest
RUNning this program and playing with it for a while, using different values
for the input prompts, you will get some amazing results. This program is
quite versatile and has many options programmed in it. Even if you have more
than enough memory, try the other graphic modes, & mix the circles in
different colors.

_   You may notice the screen go blank after you input COLOR, that is
because all the screen, sound, and I/O interrupts have been masked out
temporarily. This running portion of the program make take a few seconds,
masking out the interrupts will cause a : 34% faster execution time!! The
method of doing this will be revealed later. NOTE: This overscan program is
not a 3-D program, it is used to correct the individual TV monitor screen
sizes, for the output in the 3-D programs to follow.

    Now, load in the next program, PYRAMID.3D

_   All of these programs were written in ATARI BASIC but they execute
suprisingly fast, program design spent many LONG hours working on speeding up
this process. When READY appears, type RUN and hit RETURN. The screen
immediately asks 'SCREEN RESOLUTION?', this is self explanatory and uses
graphics modes 5, 7, & 8..If you enter a '1', you chose GRAPHICS 5 (80 x 40).
If you entered in a '2', you chose GRAPHICS 7 (160 x 80). If you chose a
'3', you get GRAPHICS 8 (320 x 160 , HI-RES). The first two modes can be
used with multi-colors. Refer to the ATARI reference manual when using the
SETCOLOR statement. Later, you will see how to get more colors in GRAPHICS
mode 8!!
GRAPHICS mode 8 is very high in resolution, and takes up 8K of memory
alone. 8K users will have to live with low resolution and programs 1 to 4
until they get more memory. Enter in a '2' for SCREEN RESOLUTION, you are now
prompted with FIELD OF VIEW. Think of having a camera with a zoom lens on it.
Small values (.1) will make the object larger as if you are 'zooming' in on
the scene. Larger values (approaching 2.0) will zoom away from the scene
making it smaller. So, enter a '.5' for FIELD OF VIEW, the software defaults
to a '.5' (.5 is about an average field of viewing).

_   The screen clears and prompts you with a VIEWERS LOCATION X,Y,Z , enter
' ?0,0,0 '. This is the location you are standing in the three coordinate
system. Next is HEADING, PITCH, & BANK, enter ' ?0,0,0 '. HEADING is the
direction your head (or your eyes) is facing.PITCH is the direction up or down
your head is aimed at.BANK is the direction you have rotated your head,
(touch your right ear to your shoulder and you will be looking at the world at
approximately a 60 degree angle). HEADING rotates on the 'Y' axis, PITCH
rotates on the 'X' axis, and BANK rotates on the 'Z' axis.

_   The following page shows actual photos of the screen so you can refer
to them, when you aren't sure where you want your scene to move to. The
numbers below them correspond to the two sets of inputs, VIEWERS LOCATION , &
HEADING PITCH, & BANK. Now, enter a '1' for COLOR. The color should always be
a '1' in the HI-RES mode. That is, unless you wish to erase a scene you just
put up.

Of course, you may wish to erase a cretain object you put on the
screen and replace it somewhere else, that can be done by re-drawing
the object you want erased using COLOR=2 or 4, then redrawing it in a
solid color (1) somewhere else. You can now have many views of an object
on the scene at once in different colors. You will learn how to rotate
the colors on the screen later, which gives a fascinating, moving
animation effect!!

   Now, enter a 'Y' for CLEAR SCREEN? YES?, and you shall see a head-on
view of a 5 sided pyramid (upside down), and hear a 'click' when each
line is drawn on the screen. This 'click' is a simple POKE 53279,0 to
'click' the internal speaker, and is used in all the other 3 three-dimen-
soinal graphics programs. Quite a handy debugging aid and nice addition.

   The screen will now say VIEWERS LOCATION X,Y,Z again, this is where
you experiment. Use the following examples as guidelines.

      NOTE: In the view of the object you have on the screen right now, you
can see only 4 of the 8 lines visible, because you are looking at it
head-on, (moving it, or your viewing location will allow you to see
the entire object.

 Note the differences in the photos below.

   The OVERSCAN correction was not used in this program so that it would
run in 8K using SCREEN RESOLUTION = '1' . It will be used in all the
rest of the 3-DIMENSIONAL GRAPHICS programs in this package.
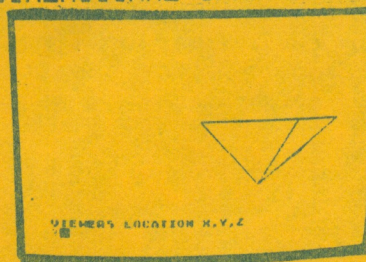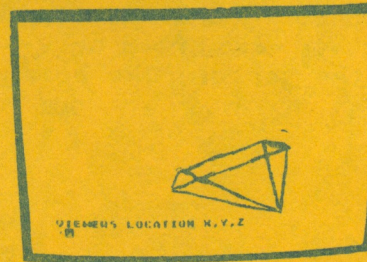
VIEWERS LOCATION	0,0,0	250,250,0
HEADING, PITCH, BANK	0,0,0	0,0,30
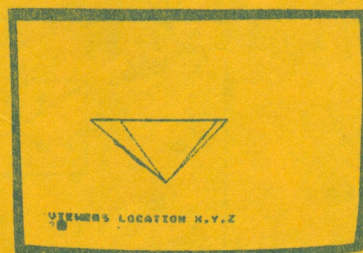
VIEWERS LOCATION	300,0,0	0,200,0	0,0,-500
HEADING, PITCH, BANK	0,0,0	0,0,0	0,0,0
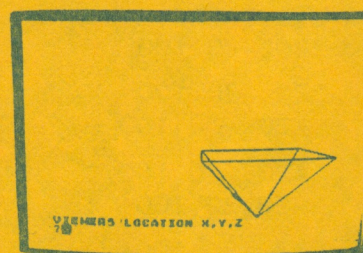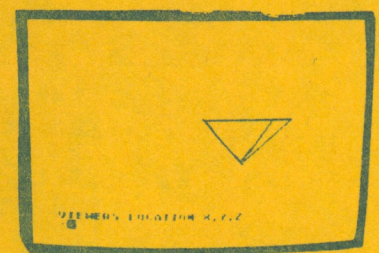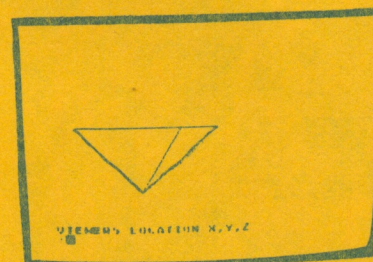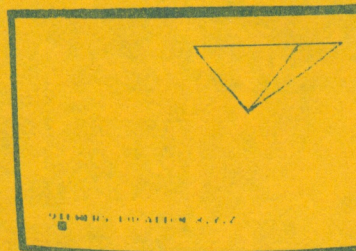
VIEWERS LOCATION	0,0,0	0,0,0	0,0,0
HEADING, PITCH, BANK	-20,0,0	0,20,0	0,0,45
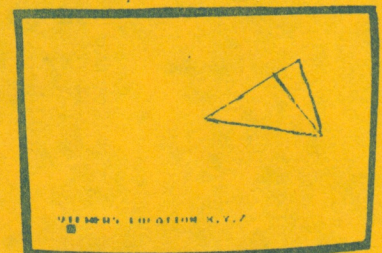
COLOR 1 , puts up a solid gold, providing you havn't changed any of
the color registers. COLOR 2 ,puts up a light green. COLOR 3 , is dark
blue. COLOR 4 is black and therefore erases anything in its path. You
have quite a compliment of color control using the SETCOLOR statement,
but with only 3 colors, you can create quite a variety of display varia-
tions, by putting 2 or 3 colors close to each other. You can draw an
object, and redraw it slightly off-center from the first one using a
different color- creating a depth effect on the object. Remember, when
using GRAPHICS 8 mode, you can only use 1 color, at least until you see
what you can do to get around that.

           PROGRAM #   , USER.3D inputting data

     After you feel you can manipulate the first 3-D program fairly enough
load in the next program, called USER.3D. This program allows the user
to input his/her own scenes through the use of simple input statements.
Try to recall your Coordinate (graph paper) geometry. After inputting the
screen resolution and field of view, enter the number of lines you want
displayed on the screen. The computer will now ask for the starting point
of line #1. Either enter in your own data, or follow this example:

                    # OF LINES-- 6

                    LINE #1 START POINT
                    ?20,20,100

                    END POINT
                    ?20,-20,100

          #2   Start Pt.   20,-20,100
               End Pt.   -20,-20,100

          #3   Start Pt. -20,-20,100
               End Pt.   -20,20,100

          #4   Start Pt. -20,20,100
               End Pt.   20,20,100

          #5   Start Pt. -20,20,100
               End Pt.   20,-20,100

          #6   Start Pt.   20,20,100
               End Pt.   -20,-20,100
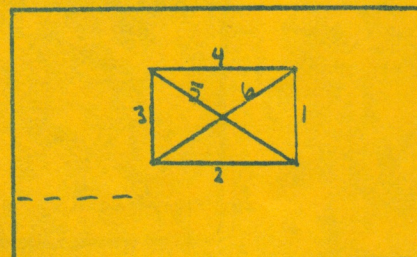
                    VIEWERS LOCATION
                    ?0,0,0
                    PITCH BANK & HEADING
                    ?0,0,0
                    COLOR
                    ?1

LETTERS '3-D'			VARIABLES IN ALPHABETICAL ORDER
1	START PT.	-30,15,100	A - LOOPING
	END PT.	-15,15,100	A1 - NESTED LOOPING WITH 'O & A'
2	START PT.	-15,15,100	B - BANK OF ROTATION
	END PT.	-15,10,100	C$(4) - INPUT 'YES' OR 'NO' STRING
3	START PT.	-15,-10,100	COL - COLOR
	END PT.	-30,-10,100	D(8) - MATRIX VAR
4	START PT.	-30,-10,100	DM - VALUES FOR INTERRUPTS
	END PT.	-30,-5,100	HZ - WIDTH OF SCREEN
5	START PT.	-30,-5,100	I - READ-IN 'Y' VAR
	END PT.	-20,-5,100	IR - INTERRUPT REGISTER VALUE
6	START PT.	-20,-5,100	J - MATRIX VAR
	END PT.	-20,0,100	L3 - TEMPORARY STORAGE- MATRIX GENERATOR
7	START PT.	-20,0,100	L4 - TEMPORARY STORAGE- MATRIX GENERATOR
	END PT.	-25,0,100	L5 - TEMPORARY STORAGE- MATRIX GENERATOR
8	START PT.	-25,0,100	L6 - TEMPORARY STORAGE- MATRIX GENERATOR
	END PT.	-25,5,100	L7 - TEMPORARY STORAGE- MATRIX GENERATOR
9	START PT.	-25,5,100	L8 - TEMPORARY STORAGE- MATRIX GENERATOR
	END PT.	-20,5,100	L9 - TEMPORARY STORAGE- MATRIX GENERATOR
10	START PT.	-20,5,100	LA - LINES OFF SCREEN WINDOW
	END PT.	-20,10,100	LN - SET TO '0' IF OFF SCREEN
11	START PT.	-20,10,100	LP - LOOPING
	END PT.	-30,10,100	LS - LENGTH OF STRINGS TO BE COLLECTED
12	START PT.	-30,10,100	M9 - HIRES MULTI-COLORS CORRECTION
	END PT.	-30,15,100	OVS - OVERSCAN CORRECTION
13	START PT.	5,0,100	P - PITCH OF VIEWER
	END PT.	-10,0,100	
14	START PT.	-10,5,100	P1 - GOTO P1 VAR
	END PT.	5,5,100	P2 - GOTO P2 VAR
15	START PT.	5,5,100	PR - LOOPING
	END PT.	5,0,100	PX - INPUT FOR VISIBLE, NON-VISIBLE PROCESSII
16	START PT.	10,15,100	PY - LOOPING
	END PT.	10,-10,100	Q(228) - 'X' DATA FOR SPACE SHUTTLE
17	START PT.	10,-10,100	R(228) - 'Y' DATA FOR SPACE SHUTTLE
	END PT.	20,-8,100	R1 - STORAGE- TRANSFORMATION MATRIX
18	START PT.	20,-8,100	R(228) - 'Y' DATA FOR SPACE SHUTTLE
	END PT.	25,-4,100	R2 - STORAGE- TRANSFORMATION MATRIX
19	START PT.	25,-4,100	R3 - STORAGE- TRANSFORMATION MATRIX
	END PT.	25,4,100	R4 - STORAGE- TRANSFORMATION MATRIX
20	START PT.	25,4,100	R5 - STORAGE- TRANSFORMATION MATRIX
	END PT.	20,8,100	R6 - STORAGE- TRANSFORMATION MATRIX
20	START PT.	20,8,100	RN - CURRENT LINE# BEING DRAWN
	END PT.	10,15,100	S - MATRIX VAR WITH 'G'
22	START PT.	15,-4,100	S(228) - 'Z' DATA FOR SPACE SHUTTLE
	END PT.	20,2,100	SN - TRIG. CORRECTION
23	START PT.	20,3,100	SR - INPUT VARIABLE
	END PT.	15,4,100	T - TRUE-FALSE STATUS (1 OR 0)
24	START PT.	15,4,100	TR - LOOPING VARIABLE
	END PT.	15,-4,100	U - READ-IN VAR FOR 'X'
25	START PT.	10,15,100	V - INPUT FOR FIELD OF VIEW
	END PT.	10,-10,100	X(5) - 'X' POINTS TO BE PLOTTED
			Y(5) - 'Y' POINTS TO BE PLOTTED
			Z(5) - 'Z' POINTS TO BE PLOTTED

VIEWERS LOCATION   -110,-30,0
HEADING,PITCH,BANK -40,-10,-45

VIEWERS LOCATION FOR
30B

When all the lines (data) have been put in, the program will ask for overscan correction, enter 'Y' or 'YES'. The preceding set of inputs will place a square with an 'X' through it on the screen. Just input your VIEWERS LOCATION, & HEADING, PITCH, & BANK as before. Try doing some rotations of this object without clearing the screen, you shall get some interesting results.

Here are some examples to use:

# of lines-- 12

CUBE

1	Start point	20,20,100
	end point	20,-20,100
2	Start point	20,-20,100
	end point	-20,-20,100
3	Start point	-20,-20,100
	end point	-20,20,100
4	Start point	-20,20,100
	end point	20,20,100
5	Start point	20,20,140
	end point	20,-20,140
6	Start point	20,-20,140
	end point	-20,-20,140
7	Start point	-20,-20,140
	end point	-20,20,140
8	Start point	-20,20,140
	end point	20,20,140
9	Start point	20,20,100
	end point	20,20,140
10	Start point	-20,20,100
	end point	-20,20,140
11	Start point	-20,-20,100
	end point	-20,-20,140
12	Start point	20,-20,100
	end point	20,-20,140

PYRAMID WITH-BASE

1	START PT.	-20,0,140
	END PT.	-20,0,100
2	START PT.	20,0,140
	END PT.	20,0,100
3	START PT.	-20,0,100
	END PT.	20,0,100
4	START PT.	-20,0,140
	END PT.	20,0,140
5	START PT.	-20,0,140
	END PT.	20,0,100
6	START PT.	20,0,140
	END PT.	-20,0,100
7	START PT.	-10,0,130
	END PT.	10,0,130
8	START PT.	-10,0,110
	END PT.	10,0,110
9	START PT.	-10,0,130
	END PT.	-10,0,110
10	START PT.	10,0,130
	END PT.	10,0,110
11	START PT.	10,0,130
	END PT.	0,50,120
12	START PT.	-10,0,130
	END PT.	0,50,120
13	START PT.	10,0,110
	END PT.	0,50,120
14	START PT.	-10,0,110
	END PT.	0,50,120

CONES

1	START PT.	0,15,100
	END PT.	0,-15,300
2	START PT.	10,5,100
	END PT.	-10,-5,300
3	START PT.	0,-15,100
	END PT.	0,15,300
4	START PT.	10,5,300
	END PT.	-10,-5,100
5	START PT.	-10,5,100
	END PT.	10,-5,300
6	START PT.	-10,5,300
	END PT.	10,-5,100
7	START PT.	0,15,100
	END PT.	-10,5,100
8	START PT.	-10,5,100
	END PT.	-10,-5,100
9	START PT.	-10,-5,100
	END PT.	0,-15,100
10	START PT.	0,-15,100
	END PT.	10,-5,100
11	START PT.	10,-5,100
	END PT.	10,5,100
12	START PT.	10,5,100
	END PT.	0,15,100
13	START PT.	0,15,300
	END PT.	-10,5,300
14	START PT.	-10,5,300
	END PT.	-10,-5,300
15	START PT.	-10,-5,300
	END PT.	0,-15,300
16	START PT.	0,-15,300
	END PT.	10,-5,300
17	START PT.	10,-5,300
	END PT.	10,5,300
18	START PT.	10,5,300
	END PT.	0,15,300

VIEWERS LOCATION   60,0,0
HEADING,PITCH,BANK 20,0,0

VIEWERS LOCATION   40,40,0
HEADING,PITCH,BANK 10,10,-2

_   With the exception of the transformation generator, clipping & pushing,
display driver, and a few other matrix routines, the program is very straight-
foreward. You will need to know how to store your own scene on memory, so
load in the next program, called CORNERS.3D . 8K users will have to refer to
program #3, the pyramid program to do what comes next. RUN the CORNERS.3D
program and enter the various values for X, Y, & Z so you can get familiar
with the scene. The scene shows what looks like a wall, a floor, and a wall
separating the two. What you might expect to see if you could see through a
corner of a room. The pattern was laid out like a grid, the surface texture
gives it depth and greatly enhances the object. Keep this method of surface -
treatment in mind when you are designing your own scene.

_   LIST the program and inspect it, or study the listing. Note that the
data for the scene starts at line # 130, and continues to line # 220.   This is
the method in which you will stare your data base. At line # 10, you will see
that LS=39. This means that there are 39 lines that will be displayed (LS=
number of lines). At line # 110, you will see the DIM statements, using Q,
R, S, as storage. Always DIM Q(A), R(A), S(A), 'A' is the total number of
start and end points, or, the number of lines multiplied by 2 plus 1. Thus,
'A' equals 79 in this particular program.

_   You may wnat to refer to the listing when doing this: In that same
line number, (# 110) fine the statement that reads: FOR TR=1 TO 78. This
FOR-NEXT loop reads the data into the Q,R,S arrays. '78' is the total number
of points to read in (LINES times 2). Change this number to the new number
of points in your data base. Do the same for PY and PR in line # 120. Then go
ahead and enter in your own data base. You can use any lines from 130 to 1999
for your data base without any modification to the program whatsoever. You
may want to try to load into DATA statements one of the examples now, or keep
reading for more information. (on everything discussed so far).

_   The listing of program #6, AUTO.3D, the last program on the cassette,
is the best one to use for study purposes. This program is intended as a DEMO
program for dealers, distributors, or just for anyone who likes to watch a
good, versatile graphics program. This listing will show the user how to
expand the base program and include 2 or more scenes.   which CAN be shown on
the screen at the same time. Load in the final program and RUN it. You will
notice many more options included in this program. The first thing you should
do is: enter in your OVERSCAN correction. The OVERSCAN should be entered
into the program at the beginning of the listing, line # 15 in this case.
NOTE: It is advised that the user make himself a copy of each program with his
OVERSCAN correction on it. Copying of this software is only allowed to the
person who purchased it.

_   Look at the 10000 block to see how you can GOSUB a certain scene and
change the perameters, (values for the scene viewing) automatically. The
3000 to 7000 block is where the Space Shuttle data is stored, and has REM
statements,   showing where the beginning of the data for X, Y, & Z are
located. When you store your data like this, start by typing a line number,
DATA, followed with the data for 'X'. For example, you want to store 3 lines,
so make LS=3. Now, take the number of lines of the value of LS, multiply it
by 2 and add 1. 3x2+1=7 . Now, DIM Q(7), R(7), and S(7).

_   NOTE: Main portions of this manual have been printed by using the BASIC
EDITOR (c) program by SEBREE'S COMPUTING. This program sells for $15.95
+$.75 P&h. Notice how nicely formatted these sections are.

There are 6 points to read in for the X,Y,& Z values, so your FOR-NEXT loops will all be from 1 to '6' . There is really nothing else which needs to be said if you just look at the listing below for a minute and see for yourself that everything just discussed has been taken care of.

Line #1 Start Pt.   -30,20,200
        End Pt.   26,15,230
Line #2 Start Pt.   -38,40,100
        End Pt.   27.5,-14,150   (Yes, full floating
Line #3 Start Pt.   0,-29.8,300   point values are
        End Pt.   -37,-43,360   permitted)

The DATA lines would be as follows:

```
1Ø GR.Ø:LS=3
11Ø DIM Q(7),R(7),S(7),X(5),Y(5),Z(5),D(8):
    FOR TR=1 TO 6:READ U:Q(TR)=U:NEXT TR
120 FOR PY=1 TO 6:READ I:R(PY)=I:NEXT PY:
    FOR PR=1 TO 6:READ E:S(PR)=E:NEXT PR
130 REM Data for X
140 DATA -30,26,-38,27.5,0,37
150 REM Data for Y
160 DATA 20,15,40,-14,-29.8,-43
170 REM Data for Z
180 DATA 200,230,100,150,300,360
```

COMPATIBILITY

If you are using this software on another system, or are planning to, there is very little that has to be changed except for the colors. Lines # 70, 80, 90 have to do with the screen resolution. HZ is the number of pixels on the screen from left to right, divided by 2 minus 1. Always specify hz on another system. GP is the graphics mode to be used, simply eliminate this. Get rid of the color command at line 2020. Insert your own commands for color control (if you have any) and anything else you need to control your graphics.

AUTO.3D DEMO program

When running program #6, you may notice a mode of AUTOMATIC LOOPING and USER INPUT. This mode determines whether or not the user will enter in his own LOCATION and the rest of the viewing information. The USER INPUT mode is the earlier default mode. The AUTO LOOPING mode is so you can have the computer choose the viewing information and demonstrate the It takes a few minutes to run through the DEMO mode, which is quite eye-catching. The program, being for display purposes, will continuously loop from control at the 10000 block. If you want to get rid of the wait loops, put a RETURN at 9998 and 9999, or leave out the gosubs at lines 6050, 8000, 8010, 8020, & 8040, and eliminate 9998 and 9999. This will show the speed of the program, it uses all the built-in trigonemetric functions for speed and high accuracy. The program runs much faster than BASIC 3-D Graphics programs on other systems.

The SPACE SHUTTLE was put into the computer by using a plastic model and drawing it out on paper, here is a copy of the blueprints for the SPACE SHUTTLE. The simple method of finding the points you are going to plot is illustrated. Say, for instance, the very center-top of the body was located at 0,0,0 , and you wish to find the values of the locations at hte tip of the nose, and the tip of the tail. You simply measure the distance along the two axis' 'A' & 'B', write down A,& B. If you wanted to DRAWTO 'C' from the tip of the tail, just measure the distance along 'A' and 'B' again, and write down your two points.   0,0,0

A

B

B

0,0,0

A

C

0

MAIN ENGINE (3)
GIMBAL TRAVEL

B

0,0,0

SPACE SHUTTLE

0,0,0

A

A

Rockwell
International
Space Division
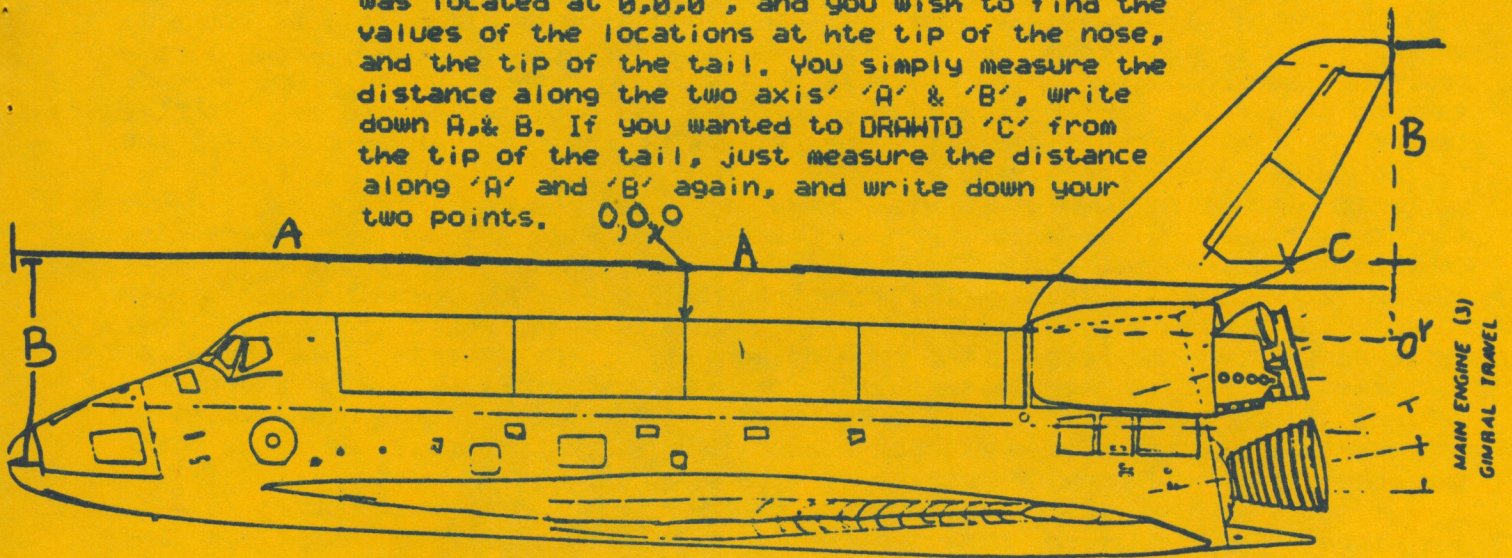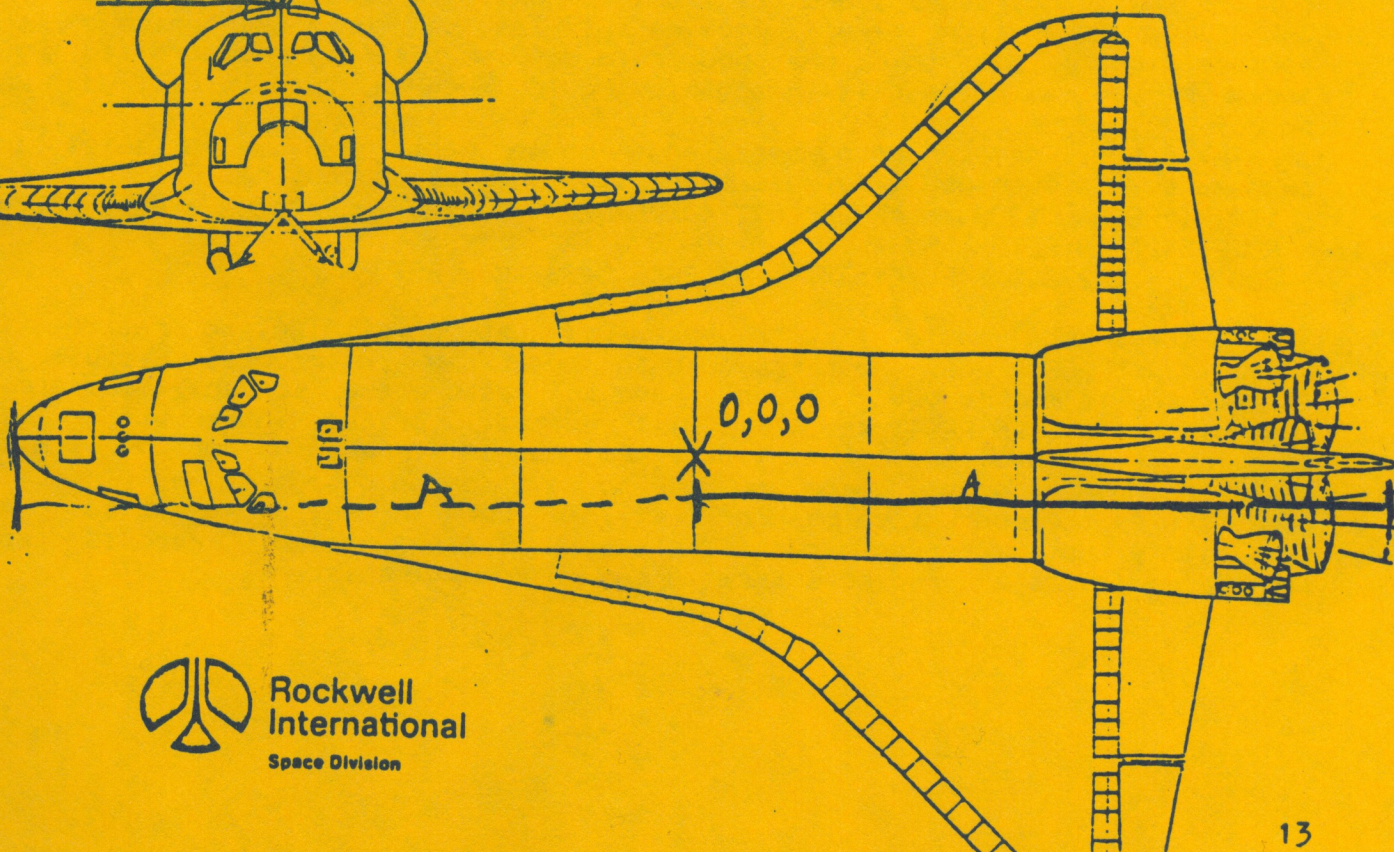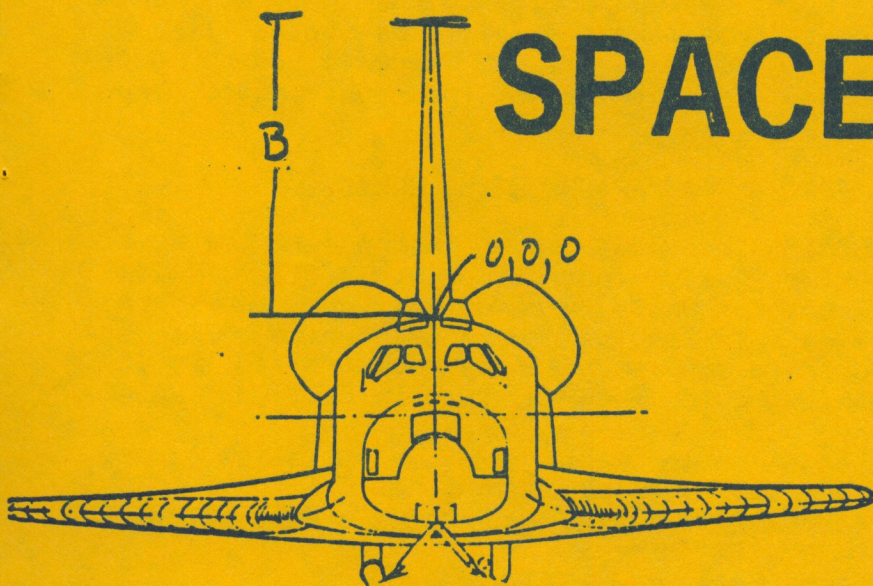
HINTS & SPECIAL TECHNIQUES

_ If youhave some trouble with the handling of your own data, first make sure you have: the same number of data numbers (e.g. DATA 10,27,32,4..) as there are 'Points you want to plot times 3'. If you would like to find out all the variables being used in any program, just type SAVE'S: , and press CONTROL and '1' to pause the listing. This will show you the entire contents of RAM in it's 'raw' form. To find the variables currently used, look at the top. The last character in each variable will be in reverse video. So, if you want to see if the variable 'R' has been used, you would expect to see it (the letter 'R') in reverse video. If you are looking for the variable 'OVS', the 'S' would be in reverse video, etc.

_ When attempting to plot curves, remember that you will sacrifice speed for quality and vice-versa. When you are doing some graphics where the computer will probably not be touched for a few minutes, I suggest to POKE out the attract mode. This is the mode the ATARI goes into when it hasn't been touched for 4 minutes. The location of the attract mode POKE is 77. You just enter POKE 77,0 somewhere in the running portion of the program. This location uses the internal counter which adds '1' to the value of PEEK(77) every few seconds. If you just poke it with a '0' once, it will just start over counting and eventually go back into attract mode. This is why you shall have to place the poke within the running portion of your program.

_ If you like the idea of the buzzer or bell going off when you like it, just type '?CHR$(253)' to ring the bell, and 'POKE 53279,0' to click the speaker. There is one other tone I like to use from the same oscillator as the bell. When the interrupts have been masked out, so the program runs faster, if also masks out the interrupts that make the bell tone sound like a warble. The result is a smooth, pleasing sound. All you have to do to get this sound is type '?CHR$(253)' when you are in the blank interrupt mode. The explanation will now follow.

MASKED OUT INTERRUPTS FOR 34% FASTER EXECUTING BASIC

_ The OVERSCAN program and the AUTO.3D programs both incorporate the use of masking out the main interrupts. The BASIC instructions to follow, will cut off the sound, screen, and the I/O interrupts. In doing so, you receive a 34% faster execution time!! Knowing this may be worth the price of this package alone!! Just think what you could do with a software-controlled BASIC speed-up! You may place this in any slow BASIC program. This is all you have to do:

_ These two lines can be numbered anything you want, and the variables can be changed to whatever you like. Use these two lines as subroutines:

```
100 MI=PEEK (54286):NQ=PEEK (53774):QC=PEEK (54272):POKE 54286,0:POKE 53774,0:
POKE 54272,0:RETURN
200 POKE 54272,QC: POKE 53774,NQ: POKE 54286,MI: RETURN
```

_ Whenever you would like to speed-up, just GOSUB 100, to restore the screen, just GOSUB 200. Just remember to reset the interrupts, because the 'BREAK' key is locked out. If you have to restore the screen immediately, just hit the SYSTEM RESET key. This will reset all of the interrupts. There is a lot of versatality in this little gem, you will probably end up using it in all of your programs, where you have to wait for an output. The first line (100) stores the current values of the interrupts in the 3 variables, and then blanks those interrupts out. The second line (200) simply recalls those storage variables and replaces them in the interrupt registers.

_   The amount of memory you need to obtain a certain view of an object depends a lot on: which graphic resolution you are using. If you have only 8K, you can only use '1' when you are asked 'SCREEN RESOLUTION?' by the software. But this is restricted to only the first, 2- 3-D programs. If you have 16K, you may use '2' when asked 'SCREEN RESOLUTION?'.But this excludes the final 3-D program. You must have at least 24K RAM to use the final program AUTO.3D. If you wish to get the SPACE SHUTTLE up in 24K RAM, you must follow these steps. First, RUN the program. When you are asked for the OVS correction, type 'YES', or 'NO', it doesn't matter which here. Next, type 'NO' to not run the DEMO mode. For SCREEN RESOLUTION, type a '2'. Type a '.5' for FIELD OF VIEW. Type a '2' for USER INPUT. Now, type a '1' for visible processing. The program will get the DATA, and clear the screen and ask you VIEWERS LOCATION? This is where you hit the BREAK key and type:
'GOTO 3000' and hit RETURN. You are now getting the data for the SPACE SHUTTLE. When VIEWERS LOCATION? comes up again, hit the 'BREAK' key again and type 'GR.7' and hit RETURN. Now type 'GOTO 2010' and hit RETURN again, and you are all set to enter a new VIEWERS LOCATION with the 'SPACE SHUTTLE' in memory. Try entering all '0's first and then changing the COLOR and moving the view elswhere. Then change the COLOR again and draw a third object in another spot on the screen. This will give a feeling of depth. NOTE: if you are using a DISK DRIVE with this software, you have to realize that you have to allow extra memory , for the DOS. If you have 32K RAM, you may do these things in HI-RESolution and get some fantastic displays. Try the following: Put up 3 (or more) different views of the object(s) you are viewing in three different colors, (COLOR 1, 2, or 3). Make sure that you do all this in GRAPHICS 7, which is 'MODE 2', when you are asked for the 'SCREEN RESOLUTION?'. Now, hit the 'BREAK' key and type 'GR.40' and hit the RETURN key.
_   You are now in HI-RES mode but you can see your scene that you built up earlier down at the bottom of the screen, in 3 COLORS!!! Now, type 'POKE 710,0'. This will change the background color to black, so that the colors are now easier to see. Now, typr 'GOTO 2010' and hit RETURN. You can now enter in new coordinates for your scenes, but note that anything you put now will be in: WHITE or BLACK (background) and one-half tha size. This is because you went to HI-RES, without changing all the other parameters. Keep building up multi-scenes without typing 'YES' when you are asked : 'CLEAR SCREEN?'. You may obtain some fascinating results in this way. You may also want to do just the opposite, go from HI-RES (with multi-scenes) to GRAPHICS 33. Then type 'GOTO 2010' (RETURN) and see what happens. Always add 32 to your current GRAPHICS MODE, to get to another mode WITHOUT clearing the screen.

WHERE ARE ALL THE COLORS LOCATED?

_   When you are in GRAPHICS 8 (or 40) [HI-RES], you may want to change the colors by hitting the 'BREAK' key, and POKEing around as follows: To change the background color, type 'POKE 710,A' . 'A' is the number of the color that you want to POKE into that color. If you want black, 'A' will be '0'. If you want white, 'A' will be '15'. You should experiment with the scale from 0 to 255, to find the color combinations you like. Color registers (locations in memory) vary from 708 to 712, depending on which GRAPHICS mode you are in. When in GRAPHICS 7 (or 39), the background will be 'POKE 712,A'. COLOR 1 will be 'POKE 708,A'. COLOR 2 will be 'POKE 709,A'. And COLOR 3 will be 'POKE 710,A', which previously (in HI-RES) controlled the background. Remember to change these when going from: MODE 2, to MODE 3 (GR.7 to GR.8).
NOTE: The advantage of POKEing these color registers rather than using the SETCOLOR statements is that: POKEs are FASTER!! And, much easier to understand than the SETCOLOR statements. They also seem to consume less memory in most instances. HAPPY POKEING AROUND!!!

```
1-3          REM STATEMENTS
5            DIM STATEMENTS
10-14        CLEAR SCREEN/RESET VARIABLES/REM ON OVERSCAN ORIGONAL VALUE
15           INPUT OF OVERSCAN (UVS) VALUE
20           INPUT AND RESET VARIABLES FOR RUNNING DEMO MODE
50           PRINT SCREEN RESOLUTION VALUES AND OPTIONS
60           SAME AS ABOVE
70           INPUT SCREEN RS
80-90        RESET VARIABLES FOR SCREEN RESOLUTION CHOSEN
100          INPUT FIELD OF VIEW
105          INPUT AUTO LOOPING or USER INPUT
106-107      INPUT NORMAL SPEED OR OPTION OF 34% FASTER SPEED/RESET VARIABLES
109-120      READ DATA STATEMENTS INTO THE Q(228), R(228), S(228) STRINGS
130-220      DATA FOR THE FIRST OBJECT TO APPEAR ON THE SCREEN
2000         RESET SCREEN FOR USER INPUT
2010-2025        USER INPUT OF LOCATION DATA, COLOR & CLEARING SCREEN MODES
2030         IF 34% FASTER SPEED CHOSEN, PEEK NEEDED VALUES
2035-2040      TRANSFORMATIONS
2050-2090        ACTUAL LOOP FOR PLOTTING THE DATA ON THE SCREEN
2095         IF 34% FASTER SPEED WAS CHOSEN, RESET INTERRUPTS
2100         RETURN FROM ORIGONAL SPOT
2500-2510        MATRIX MULJIPLIER LOOP
2600-2626        LINE PUSHING AND CLIPPING MATRIX
2632-2640        CLIPPING CONTINUED/ CHECK FOR LINE OFF SCREEN
2645-2648        STARTING POINT FOR LINE THAT FALLS OFF THE SCREEN
2655-2658        SAME AS ABOVE BUT FOR END POINT OF LINE
2660-2680        ON OR OFF SCREEN CHECK GOTO's
2694-2706        WHICH DIRECTION TO PUSH THE LINE
2715-2720        PUSH THE POINT TO THE LEFT
2725-2730        PUSH THE POINT TO THE RIGHT
2742-2745        PUSH THE POINT UP
2756-2760        PUSH THE POINT DOWN
2800-2960        SECTION FOR PROJECTING THE LINE ONTO THE SCREEN
3000-3025        READ-IN THE DATA FOR THE SPACE SHUTTLE
3030-4000        DATA FOR X  Q(228)
4010-4090        DATA FOR Y  R(228)
5000-6000        DATA FOR Z  S(228)
6010-6045        DATA FOR THE FOUR VIEWS OF THE SPACE SHUTTLE
6050     RING BELL WHEN FINISHED
6060-6095        STATEMENTS GIVEN AT ENDING OF AUTO LOOP
7000-7995        GOSUB DISPLAY SUBROUTINE/RESET SCREEN/READ IN INTERRUPTS
8000-8060        PRINT DATA FOR AOTO LOOPING
9000-9050        DISPLAY SUBROUTINE
9998-9999        WAIT SUBROUTINES
10000-10060      EACH LINE HAS SEPERATE DATA FOR VIEWS OF OBJECTS IN AUTO MODE
11000-11030      DRAW BORDER FOR OVERSCAN CORRECTION SUBROUTINE
12000-13000      ROTATE COLORS SUBROUTINE
```

_   NOTE: Main portions of this manual have been printed by using the
'BASIC EDITOR (c)/ Mini- Word Processor' program by SEBREE'S COMPUTING. This
program sells for $15.95 +$.75 p&h. Notice how nicely formatted these sections
are.

ROTATION OF COLORS

Here is the subroutine used in the DEMO AUTO.3D program to rotate the
colors.

```
1200 ? ROTATE the COLORS of your scene in anycombination you want!!!
12010 FOR TR=0 to 30:SETCOLOR 2,8,2:SETCOLOR 1,8,5:SETCOLOR 0,8,8
GOSUB 13000:SETCOLOR 0,8,2:SETCOLOR 2,8,5
12020 SETCOLOR 1,8,8:GOSUB 13000:SETCOLOR 1,8,2:SETCOLOR 0,8,5
SETCOLOR 2,8,8:GOSUB 13000:NEXT TR
12030 POKE 708,40:POKE 709,202:POKE 710,148:POKE 711,70:POKE 712,0:RETURN
13000 FOR PY=0 TO 19:NEXT PY:RETURN
```

You can alter the total effect by changing the colors just a little bi.
Whenever you would like to have an option to rotate your colors, just use
this subroutine, and GOSUB 12000.
Try putting up a scene in GRAPHICS 7 , SCREEN RESOLUTION mode 2, put the
scene in all three colors. Now rotate them, and see the wonderful effect
you get.

HEAVY TOPIC- MULTI COLORS IN HI-RES MODE

The following program is ont on the tape, it is suggested for those
of you who are interested in getting more colors in HI-RES mode than
the DEFAULT two (foreground & background). It also shows a method of
moving your entire graphics page down the screen, to allow for printed
material at the top (in 3 colors), and graphics from the center of the
screen to the bottom.
It is highly suggested that you study this program and chapter.
Listing
```
2 REM RANGE IS XMAX=159 : YMAX=97
3 REM POKE87,6 ---PRODUCES A DUPLICATE IMAGE ON THE RIGHT & LEFT SIDES
4 REM POKE 87,9 ---GIVES XMAX=159 , YMAX=159 COLOR (0-8)GIVES DIFFERENT COLORS &
TYPES OF LINES
5 TRAP 120
10 OPEN #3,4,0,"S:"
20 GRAPHICS 8
30 SETCOLOR 2,0,0:POKE 752,3
40 Y=PEEK(561)*256+PEEK(560):FOR X=0 TO 2:POKE Y+136+X,PEEK(Y+199+X):NEXT X
50 II=Y+6:POKE II-3,71:POKE II,7:FOR J=1 TO 6:POKE II+J,6:NEXT J:POKE 87,1
60 ? #6;"ThE 9RaPhiC PaGE HaS bEEn mOVEd DoWnWaRd To AlLoW PRintEd   LeTTErS aT
tHE tOP  Of THE ScREEn ";
61 ? #6;"In-    StEaD of tHE BoTTom.";
62 ? #6;"     graphics:"
63 ? #6;" HIRES In 4 CoLoRs."
64 GOTO 100
65 POKE 87,7
70 COLOR 2:PLOT 35,6:FOR K=1 TO 10:DRAWTO RND(1)*159,RND(1)*80+7:NEXT K
80 POKE 87,7:COLOR 3:PLOT 35,6:FOR K=0 TO 10:DRAWTO RND(1)*159,RND(1)*80+7:NEXT
K
100 POKE 710,0:FOR A=0 TO 159:POKE 87,7:COLOR RND(1)*5:PLOT A,6:DRAWTO A,RND(1)*
80+7:NEXT A
110 ? CHR$(253):GOTO 70
120 ? CHR$(253):? CHR$(253)
130 GOTO 70
```

Note the upper and lower case letters in lines 60, & 61.
Lines 40 & 50 provide the necessary adjustments for moving the graphics
page of memory into this specific location.

Lines 60-63 do the colored printing at the top of the screen. Lines 65 to
100, take care of the graphics.
The secret is in the manipulation of POKE 87. This location in memory
has all kinds of odd curiosities about it.
In order to get more colors in HI-RES, you have to fool the ATARI into
thinking it is in GRAPHICS 7 mode. You do this by POKEing the location '87'
with the value of the graphics mode you are in (assuming GRAPHICS 8) minus
one. So PEEK(87) will return an '8' if you are in GRAPHICS 8, this is where
we put POKE 87,7. Whichever Graphics mode you are in, the POKE 87,N will
change the dimension of the screen. Here is a program listing which
has a breakdown of the screen dimensions corresponding to the value in
which 87 was POKEd.

```
1 REM LIMITS
5 GRAPHICS 0:? :?
10 ? "POKE 87,1  = XMAX=19 , YMAX=23"
20 ? "POKE 87,3  = XMAX=39 , YMAX=23"
30 ? "POKE 87,5  = XMAX=39 , YMAX=47"
40 ? "POKE 87,7  = XMAX=159 , YMAX=95"
50 ? "POKE 87,9  = XMAX=79 , YMAX=191"
```

The above program assumes you are currently in GRAPHICS 8, when you POKE these
values
Here is a MOIRE program listing which uses all of these graphics modes

```
1 REM MOIRE
10 MX=79:MY=191:PO=9
20 GOSUB 1000:GOSUB 2000
30 MX=159:MY=95:PO=7
40 GOSUB 1000:GOSUB 2000
50 MX=79:MY=47:PO=5
60 GOSUB 1000:GOSUB 2000
70 MX=39:MY=23:PO=3
80 GOSUB 1000:GOSUB 2000
90 MX=19:MY=23:PO=1
100 GOSUB 1000:GOSUB 2000
990 END
1000 REM
1010 GRAPHICS 24:POKE 710,0
1020 FOR Y=MY TO 0 STEP -1:Q=Y:GOSUB 1400:PLOT 0,Y:DRAWTO MX/2,MY:NEXT Y
1100 FOR X=0 TO MX:Q=X:GOSUB 1400:PLOT X,0:DRAWTO MX/2,MY:NEXT X
1220 FOR Y=0 TO MY:Q=Y:GOSUB 1400:PLOT MX,Y:DRAWTO MX/2+1,MY:NEXT Y:RETURN
1300 FOR X=MX TO 0 STEP -1:Q=X:GOSUB 1400:PLOT X,MY:DRAWTO MX/2,MY/2:NEXT X:RETU
RN
1370 RETURN
1400 IF Q=125 THEN Q=13
1410 IF Q=155 THEN Q=11
1420 POKE 87,PO:COLOR Q
1430 RETURN
2000 FOR AA=0 TO 1000:NEXT AA:RETURN
```

VERY IMPORTANT NOTE: If you want to move back and forth between modes and
still see what was on the screen, in other words, without erasing the
screen, just use a GRAPHICS 40, in place of GRAPHICS 8 (ADD 32 to the
graphics mode you want to goto). The ATARI will open-up the screen without
erasing what was on it !!!

Just follow the examples in the above programs to get your multi-colors on
the screen.

When you POKE 87,7 , you will only have the multicolors from the top to the
center of the screen. Unless you POKE another value, you will have to make sure
that no lines go beyond the center of the screen. There is a way around this as
demonstrated in our scribbling program. The software checks to see if the line
is below the center of the screen, if it is, the POKE 87,7 is changed back to
an '8' and the COLOR is made COLOR 1 (white) and the program continues. When
your joystick is above the center of the screen, you have the option of pressing
1,2,3,or 4 on the keyboard to change the color that you are drawing in.
Here is a listing of the SCRIBBLING program, note:none of the programs after
AUTO.3D are on tape, these are user-experimental programs.
NOTE: SCRIBBLER REQUIRES ONE JOYSTICK.

```
10 GRAPHICS 0:? "        HIRES SCRIBBLER":GOSUB 600
20 POKE 764,255:? :? "      PRESS  RETURN  TO START"
30 IF PEEK(764)=255 THEN 30
90 OPEN #3,4,0,"S:":B=1
100 GRAPHICS 24:POKE 710,0:POKE 764,255:X=159:Y=100:POKE 87,8
120 JS=STICK(0)
130 POKE 764,255:IF JS=7 OR JS=6 OR JS=5 THEN X=X+1
150 IF JS=11 OR JS=10 OR JS=9 THEN X=X-1:IF X<1 THEN X=1
170 IF JS=14 OR JS=6 OR JS=10 THEN Y=Y-1:IF Y<1 THEN Y=1
180 IF Y<95 THEN POKE 87,7
190 IF JS=13 OR JS=9 OR JS=5 THEN Y=Y+1:IF Y>94 THEN POKE 87,8
200 IF PEEK(764)<>255 THEN 400
210 IF JS=15 THEN 500
220 IF STRIG(0)=0 THEN 250
230 IF PEEK(87)=7 THEN COLOR B:PLOT X/2-1,Y:FOR A2=1 TO 5:NEXT A2:COLOR 0:PLOT X
/2-1,Y:GOTO 120
240 COLOR 1:PLOT X,Y:FOR A2=1 TO 5:NEXT A2:COLOR 0:PLOT X,Y:GOTO 120
250 COLOR B:IF PEEK(87)=7 THEN PLOT X/2-1,Y:GOTO 120
260 COLOR 1:PLOT X,Y:GOTO 120
400 P=PEEK(764):IF P=31 THEN B=1
410 IF P=30 THEN B=2
420 IF P=26 THEN B=3
430 IF P=24 THEN B=4
440 IF P=118 OR P=54 THEN 100
450 COLOR B:GOTO 210
500 IF PEEK(87)=7 THEN COLOR B:PLOT X/2-1,Y:COLOR 0:PLOT X/2-1,Y:GOTO 120
510 COLOR 1:PLOT X,Y:COLOR 0:PLOT X,Y:GOTO 120
600 ? " REQUIRES 1 JOYSTICK CONTROLLER."
610 ? " JUST PRESS TRIGGER TO DRAW A SOLID   LINE, RELEASING TRIGGER WILL LEAVE
A"
620 ? "BLINKING DOT AT THE LOCATION YOU ARE   CURRENTLY AT."
630 ? " MOVE JOYSTICK ANY DIRECTION TO MOVE   YOUR DRAWING CURSOR."
640 ? " FROM TOP TO CENTER, YOU HAVE 4 COLORSTO DRAW WITH, JUST PRESS 1 THROUGH
4  ON THE KEYBOARD AND WAIT 2 SEC-"
650 ? "ONDS BEFORE RESUMING YOUR SCRIBBLING."
655 ? " NOTE: COLOR 4 IS BLACK (BACKGROUND)"
660 ? " FROM THE CENTER OF THE SCREEN TO THE BOTTOM, YOU HAVE JUST BLACK AND WHI
TE TO DRAW WITH."
670 ? " PRESS THE CLEAR KEY TO CLEAR SCREEN"
700 RETURN
```

   The next two programs are variations of each other, they give an example of the
modes type of graphics.

19

```
0 REM DOUBLES
1 REM RANDOMLY MOVING VECTOR GENERATOR
2 REM BY Tim Hays
3 SETCOLOR 4,5,4:R=0
4 POKE 752,3
5 IF RND(0)*2>1 THEN 70
6 GOTO 70
10 GRAPHICS 8
20 HX=159:HY=95
60 GOTO 80
65 REM
70 GRAPHICS 8:SETCOLOR 2,0,0:R=1:COLOR 1:HX=79:HY=47:POKE 87,5
80 FOR D=1 TO 2
90 A=10*RND(1)-5:B=10*RND(1)-5:A1=10*RND(1)-5:B1=10*RND(1)-5
100 X=HX*RND(1):Y=HY*RND(1)
110 X1=HX*RND(1):Y1=HY*RND(1)
120 COLOR RND(0)*16:T=RND(0)*2
125 FOR C=1 TO 16-(ABS(R-1)*14)
130 IF X>HX OR X<1 OR Y>HY OR Y<1 THEN POP :GOTO 90
140 IF X1>HX OR X1<1 OR Y1>HY OR Y1<1 THEN POP :GOTO 90
145 IF T>1 THEN COLOR RND(8)
150 PLOT X,Y:DRAWTO X1,Y1
160 X=X+A:Y=Y+B:X1=X1+A1:Y1=Y1+B1
170 NEXT C
180 NEXT D
190 RUN
END

0 REM VIDEO2.ART
1 REM RANDOMLY MOVING VECTOR GENERATOR
2 REM BY Tim Hays
3 SETCOLOR 4,5,4:R=0
4 POKE 752,3
5 IF RND(0)*2>1 THEN 70
6 GOTO 70
10 GRAPHICS 8
20 HX=159:HY=95
60 GOTO 80
65 REM
70 GRAPHICS 24:SETCOLOR 2,0,0:R=1:COLOR 1:HX=79:HY=191:POKE 87,9
80 FOR D=1 TO 3
90 A=10*RND(1)-5:B=10*RND(1)-5:A1=10*RND(1)-5:B1=10*RND(1)-5
100 X=HX*RND(1):Y=HY*RND(1)
110 X1=HX*RND(1):Y1=HY*RND(1)
120 COLOR RND(0)*16:T=RND(0)*2
125 FOR C=1 TO 30-(ABS(R-1)*14)
130 IF X>HX OR X<1 OR Y>HY OR Y<1 THEN POP :GOTO 90
140 IF X1>HX OR X1<1 OR Y1>HY OR Y1<1 THEN POP :GOTO 90
145 IF T>1 THEN COLOR RND(8)
150 PLOT X,Y:DRAWTO X1,Y1
160 X=X+A:Y=Y+B:X1=X1+A1:Y1=Y1+B1
170 NEXT C
180 NEXT D
190 RUN
END
```

Notice the colors and the types of lines being generated.

MIXING TEXT MODES

The very first program, called DISPLAY, had a rather unusual text mode setup
to it. This is not hard to learn how to do. The best way to learn about any
type of graphics functions on this machine, is to experiment with it.
Inside the ATARI lies a set of numbers telling it which graphics mode it is in.
This list of numbers change with each new declaration of a graphics mode.
There is one number for each graphics row on the screen. Ingraphics 7+16, there
are 96 rows, (96 numbers), these numbers in the list control what graphics mode
each row is being displayed in. So it is a simple matter of changing these num-
bers in the list the correct way to achieve what you want. The only problem is
that our new mode has to have a total of 192 scan lines. Exactly.

MODE	0	1	2	3	4	5	6	7	8
NUMBER	66	70	71	72	73	74	75	77	79

Look at the table above. All you have to do is add-up any set of modes which
the the NUMBERs total 192.
You will have to PEEK into the computer's memory for two locations to tell you
where the list of numbers is located.
```
10 A=PEEK (560)+PEEK(561)*254+4
```

The next thing to do is to make a correction if the mode you want on the top of
the screen is not the same as the one which consumes the most memory.
```
20 POKE A-1,71
```

Go down the list and change any numbers which need correction to get your mode.
These lines are just examples...
```
30 POKE a+2,7:POKE A+3,7
```

Numbers for mode 1 don't need to be changed, they are the default. Line 30 just
took care of mode 2.
```
40 FOR B=6 TO 21:POKE A+B,8:NEXTB
```

That just took care of the numbers being corrected for mode 3.
```
50 POKE A+22,65:POKE A+23,PEEK(560):POKE A+24, PEEK(561)
```

Tell the ATARI to go back to the beginning of the list.
When designing your own modes, the only changes to be made in line 50 are the
numbers 22, 23, 24. these are the 3 row numbers after the last row you used on

the screen. HAVE FUN!!!!

LISTING OF DISPLAY PROGRAM #1

```
4 REM DISPLAY TITLES PROG. FOR 3D GRAPHICS PAC. COPYRIGHT 1980 SEBREE'S
10 GRAPHICS 0:POKE 752,1:J=6+PEEK(742)*256+PEEK(741)
20 SETCOLOR 2,2,4:SETCOLOR 1,2,10:SETCOLOR 0,2,10:SETCOLOR 4,2,4:SETCOLOR 3,6,10
30 POSITION 1,2:? "SEBREE'S COMPUTING":POSITION 26,5:? "PRESENTS"
40 POSITION 4,7:? "3-DIMENSIONAL     COMPUTER GRAPHICS"
50 POSITION 3,17:? "(C)'80 Sebree's Computing-ATARI 800"
60 POKE J+2,7:POKE J+6,6:POKE J+8,6:POKE J+9,7
70 POKE 82,2:POKE 752,0
80 ? CHR$(253):FOR A=1 TO 28
100 POSITION A,9:? "
```

```
110 POSITION A,10:? "            "
120 POSITION A,11:? "
```

SPECIAL GRAPHICS CHARACTERS

```
130 POSITION A,12:? "            "
140 POSITION A,13:? "            "
150 POSITION A,14:? "            "
160 POSITION A,15:? "
```

```
170 POSITION A,16:? " ":POKE 53279,0
200 POKE 755,4:FOR B=0 TO 40:NEXT B:POKE 755,0:FOR B=0 TO 32:NEXT B:NEXT A
280 FOR A=1 TO 13:POSITION 1,9:? "":NEXT A
290 POSITION 3,17:? "(C)'80 Sebree's Computing-ATARI 800"
300 POSITION 8,9:? "Programs in this package:":POSITION 4,11:? "1) DISPLAY Title
s 2) Overscan"
310 ? :? "  3) PYRAMID.3D      4) USER.3D":? :? "  5) CORNERS.3D     5) AUTO.3D
 demo":POKE 755,2:CLOAD
```

L-1

LISTING OF OVERSCAN CORRECTION PROGRAM #2

```
1 REM CIRCLE (POLYGON) PLOTTER:PROG.2
2 REM OVERSCAN prog. for 3D Graphics  Package, Version 8 , 10/2/80
3 REM COPYRIGHT 1980 SEBREE'S : PROG. BY TIM HAYS : ALL RIGHTS RESERVED
4 HZ=159:GRAPHICS 0:POKE 710,0:DIM Y$(3):?
5 ? " CIRCLE (POLYGON) PLOTTER."
10 ? " DRAWS ALL CHORDS WITHIN AN N-GON."
20 ? " USED TO DETERMINE OVERSCAN OF THE     MONITOR SO A CIRCLE IS NOT AN ELIPSE
   AND A SQUARE IS NOT A RECTANGLE."
30 ? :? "USER INPUTS THE NUMBER OF SIDES OF ANYPOLYGON WITH AN EVEN NUMBER OF SI
DES."
40 ? " MANY-SIDED POLYGONS ARE CLASSIFIED ASCIRCLES, AND WILL LOOK CIRCULAR  IN
    HIRES.":?
50 ? " ENTER EVEN NUMBER OF SIDES ONLY."
60 TRAP 60:? CHR$(253):? " NUMBER OF SIDES?":INPUT N:DIM B(N),C(N),H(N)
80 TRAP 80:? " RADIUS OF PLOT? (HI-RES MAX=154)":INPUT R
85 ? " DIAMETER PLOT ONLY? 'Y'=YES":INPUT Y$:IF Y$(1,1)="Y" THEN CIR=1
86 IF CIR<>1 THEN ? "FANTAIL DISPLAY? 'Y'=YES, 'N'=SPOKES":INPUT Y$:IF Y$(1,1)="
Y" THEN FAN=1
90 P1=6.2831853/N:FLAG=0:? " GRAPHICS RESOLUTION? (MODE 3-8)":TRAP 90:INPUT GR
92 TRAP 50000
93 OP=32:? " CLEAR THE SCREEN? 'Y'=YES":INPUT Y$:IF Y$(1,1)="Y" THEN OP=0
94 TRAP 95:GRAPHICS GR+OP
95 FOR A=0 TO 3:? "                                   ":NEXT A:? " COLOR? ALW
AYS '1' FOR HIRES":INPUT CL:COLOR CL
96 NM=PEEK(54286):IR=PEEK(53774):DM=PEEK(54272):POKE 54272,0:POKE 53774,0:POKE 5
4286,0:IF GR=8 THEN POKE 710,0
98 TRAP 50000
100 OVS=0.577:REM THIS IS WHERE THE OVERSCAN VARIABLE IS MANIPULATED.     THE ORIG
ONAL OVS WAS .577
110 REM THE OVERSCAN VARIABLE ONLY  CONTROLS THE 'X' DIRECTION. MAKING  OVS SMAL
LER WILL SHORTEN THE WIDTH.
120 REM MAKING OVS LARGER THAN 1.07    WILL MAKE THE FIGURE WIDER.
125 OVS=OVS*0.9272097
130 FOR I=1 TO N:B(I)=R+10+R*COS((I-1)*P1):C(I)=R+10+R*SIN((I-1)*P1):NEXT I
135 ? CHR$(253):POKE 54272,DM:POKE 53774,IR:POKE 54286,NM
140 FOR I=1 TO INT(N/2):FOR J=1 TO N:H(J)=0:NEXT J
190 IF I<>N/2 THEN 240
200 FOR J=N/2+1 TO N:H(J)=1:NEXT J
240 J2=1:F=1
250 N1=1
260 IF I=N/2 THEN N1=I
295 IF FLAG=0 THEN PLOT OVS*(B(J2)+HZ)-OVS*HZ,C(J2)/2:SV1=B(J2):SV2=C(J2)
300 FOR J=1 TO N1:J1=J2
320 IF H(J1)=0 THEN 350
330 J1=J1+1:IF J1>N THEN J1=J1-N
340 F=1:GOTO 320
350 J2=J1+I:IF J2>N THEN J2=J2-N
360 J2=J1+I
380 H(J1)=N
400 IF CIR=1 THEN 410
402 IF F=1 THEN PLOT OVS*(B(J1)+HZ)-OVS*HZ,C(J1)/2
410 IF FAN=1 THEN PLOT OVS*(SV1+HZ)-OVS*HZ,SV2/2
420 F=0:FLAG=FLAG+1:DRAWTO OVS*(B(J2)+HZ)-OVS*HZ,C(J2)/2:POKE 53279,0
430 NEXT J:NEXT I:DRAWTO OVS*(SV1+HZ0-OVS*HZ,SV2/2:CLR :DIM Y$(3):GOTO 60
END.
```

L-2

LISTING OF PYRAMID.3D PROGRAM #3

```
1 REM PYRAMID.3D Program version 3
2 REM Copyright 1980 SEBREE'S:T.HAYS :ALL RIGHTS RESERVED
10 GRAPHICS 0:LS=8:POKE 710,0
50 POSITION 5,4:? "Screen Resolution?":POSITION 11,5:? "X , Y":POSITION 5,6:?
   "1)  80 x 40"
60 POSITION 5,7:? "2) 160 x 80":POSITION 5,8:? "3) 320 x 160 HIRES"
70 INPUT SR:IF SR=1 THEN HZ=39:GP=5:GOTO 100
80 IF SR=2 THEN HZ=79:GP=7:GOTO 100
90 HZ=159:GP=8
100 POSITION 4,11:? "Field of VIEW  .1 Wide--2.0 Small":INPUT V
110 DIM Q(16),R(16),S(16),X(5),Y(5),Z(5),D(8),C$(3):FOR TR=1 TO 16:READ U:Q(TR)=
   U:NEXT TR
120 FOR PY=1 TO 16:READ I:R(PY)=I:NEXT PY:FOR PR=1 TO 16:READ E:S(PR)=E:NEXT PR
130 DATA 0,400,400,400,400,0,0,0,0,200,200,0,200,400,400,200
140 DATA 0,0,0,0,0,0,0,0,0,0,400,400,0,400,0,0,400
150 DATA 1000,1000,1000,1400,1400,1400,1400,1000,1000,1200,1200,1400,1200,1000,1
   400,1200
2000 GRAPHICS GP:SN=0.017453292:POKE 710,0:LA=0:IF GP<>8 THEN POKE 710,148
2010 TRAP 2010:? "VIEWERS LOCATION X,Y,Z":INPUT U,I,E:X(3)=-U:Y(3)=I:Z(3)=-E:TRA
   P 2020
2020 ? "Heading, Pitch, & Bank (X,Y,Z)":INPUT H,P,B:P=-P:? "COLOR":INPUT COL:H=-
   H:COLOR COL:IF ABS(B)>89 THEN 2020
2025 TRAP 2025:? "CLEAR Screen? YES?":INPUT C$:IF C$(1,1)="Y" THEN GRAPHICS GP:I
   F GP=8 THEN POKE 710,0
2030 R6=COS(H*SN):R5=SIN(H*SN):R4=COS(B*SN):R3=SIN(B*SN):R2=COS(P*SN):R1=SIN(P*S
   N):L1=R4*R6+R5*R1*R3
2040 L2=-R6*R3+R5*R1*R4:L3=R5*R2*V:L4=R2*R3:L5=R2*R4:L6=-R1*V:L7=-R5*R4+R6*R1*R3
   :L8=R5*R3+R1*R4:L9=R2*R6*V
2050 FOR LP=1 TO LS*2-1 STEP 2:X(1)=Q(LP):Y(1)=R(LP):Z(1)=S(LP)
2060 X(5)=Q(LP+1):Y(5)=R(LP+1):Z(5)=S(LP+1):GOSUB 2500
2070 IF LN=0 THEN LA=LA+1:? "LINE off screen window=";LA:GOTO 2090
2080 PLOT X(2)+HZ,(Y(2)+HZ)/2:DRAWTO X(4)+HZ,(Y(4)+HZ)/2
2090 POKE 53279,0:NEXT LP:GOTO 2010
2500 FOR A=1 TO 5 STEP 4:G=X(A)+X(3):S=Y(A)+Y(3):J=Z(A)+Z(3):X(A)=G*L1+S*L4+J*L7
2510 Y(A)=G*L2+S*L5+J*L8:Z(A)=G*L3+S*L6+J*L9:NEXT A
2600 FOR A=1 TO 5 STEP 4:D(A)=0:D(A+1)=0:D(A+2)=0:D(A+3)=0
2610 IF X(A)<-Z(A) THEN D(A)=1
2620 IF X(A)>Z(A) THEN D(A+1)=1
2622 IF Y(A)<-Z(A) THEN D(A+2)=1
2623 IF Y(A)>Z(A) THEN D(A+3)=1
2626 NEXT A
2632 FOR A=1 TO 4:IF D(A)=0 THEN 2640
2636 IF D(A)=D(A+4) THEN 2668
2640 NEXT A
2645 FOR A=1 TO 4:IF D(A)=1 THEN 2675
2648 NEXT A
2655 FOR A=5 TO 8:IF D(A)=1 THEN 2680
2658 NEXT A
2660 LN=1:GOTO 2800
2668 LN=0
2670 RETURN
```

Prog. 3 cont.

```
2675 A=1:S=5:GOTO 2694
2680 A=5:S=1
2694 IF D(A)=1 THEN 2725
2696 IF D(A+1)=1 THEN 2715
2698 IF D(A+2)=1 THEN 2742
2700 IF D(A+3)=1 THEN 2756
2706 GOTO 2660
2715 J=(Z(A)-X(A))/(X(S)-X(A)-Z(S)+Z(A))
2720 X(A)=J*(Z(S)-Z(A))+Z(A):Y(A)=J*(Y(S)-Y(A))+Y(A):Z(A)=X(A):GOTO 2600
2725 J=(Z(A)+X(A))/(X(A)-X(S)-Z(S)+Z(A))
2730 X(A)=J*(Z(A)-Z(S))-Z(A):Y(A)=J*(Y(S)-Y(A))+Y(A):Z(A)=-X(A):GOTO 2600
2742 J=(Z(A)+Y(A))/(Y(A)-Y(S)-Z(S)+Z(A))
2745 X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(A)-Z(S))-Z(A):Z(A)=-Y(A):GOTO 2600
2756 J=(Z(A)-Y(A))/(Y(S)-Y(A)-Z(S)+Z(A))
2760 X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(S)-Z(A))+Z(A):Z(A)=Y(A):GOTO 2600
2800 IF Z(1)=0 THEN Z(1)=1.0E-03
2850 IF Z(5)=0 THEN Z(5)=1.0E-03
2860 X(2)=X(1)/Z(1)*HZ:Y(2)=Y(1)/Z(1)*HZ:X(4)=X(5)/Z(5)*HZ:Y(4)=Y(5)/Z(5)*HZ:RET
URN
END
```

LISTING OF USER.3D PROGRAM #4

```
1 REM USER.3D Program version 3
2 REM Copyright 1980 SEBREE'S:T.HAYS: All Rights Reserved
10 GRAPHICS.0:POKE 710,0
50 POSITION 5,4:? "Screen Resolution?":POSITION 11,5:? "X  ,   Y":POSITION 5,6:?
"1)   80  x  40"
60 POSITION 5,7:? "2)  160  x  80":POSITION 5,8:? "3)  320  x  160  HIRES"
70 TRAP 70:INPUT SR:IF SR=1 THEN HZ=39:GP=5:GOTO 100
80 IF SR=2 THEN HZ=79:GP=7:GOTO 100
90 HZ=159:GP=8
100 TRAP 100:POSITION 4,11:? "Field of VIEW  .1 WIDE--2.0 SMALL":INPUT V
110 TRAP 110:? "HOW MANY LINE SEGMENTS?":INPUT LS:DIM Q(LS*2+1),R(LS*2+1),S(LS*2
+1),X(5),Y(5),Z(5),D(8),C$(3)
115 GRAPHICS 0:? :? "ENTER DATA IN THE FORM OF 'X,Y,Z'.":POKE 710,0
120 FOR LP=1 TO LS:? :? "`````````````````````````````````````````":? "LINE # ";LP;
" START PT(X,Y,Z)"
125 TRAP 125:INPUT U,I,E:I=-I
130 TRAP 130:Q(LP*2-1)=U:R(LP*2-1)=I:S(LP*2-1)=E:? :? "END POINT":INPUT U,I,E:I=
-I
140 Q(LP*2)=U:R(LP*2)=I:S(LP*2)=E:NEXT LP
200 OVS=0:? "Want OVERSCAN correction (OVS)? YES?":INPUT C$:IF C$(1,1)="Y" THEN
OVS=0.577
2000 GRAPHICS GP:SN=0.017453292:IF GP=8 THEN POKE 710,0:LA=0:IF OVS THEN GOSUB 1
1000
2010 TRAP 2010:? CHR$(253):? "VIEWERS LOCATION X,Y,Z":INPUT U,I,E:X(3)=-U:Y(3)=I
:Z(3)=-E
2020 TRAP 2020:? "Heading, Pitch, & Bank (X,Y,Z)":INPUT H,P,B:? "COLOR":H=-H:P=-
P:INPUT COL:COLOR COL:IF ABS(B)>89 THEN 2020
2025 TRAP 2025:? "CLEAR Screen? YES?":INPUT C$:IF C$(1,1)="Y" THEN GRAPHICS GP:I
F GP=8 THEN POKE 710,0:IF OVS THEN GOSUB 11000
2030 R6=COS(H*SN):R5=SIN(H*SN):R4=COS(B*SN):R3=SIN(B*SN):R2=COS(P*SN):R1=SIN(P*S
N):L1=R4*R6+R5*R1*R3
```

```
2040 L2=-R6*R3+R5*R1*R4:L3=R5*R2*U:L4=R2*R3:L5=R2*R4:L6=-R1*U:L7=-R5*R4+R6*R1*R3
•L8=R5*R3+R1*R4:L9=R2*R6*U
2050 FOR LP=1 TO LS*2-1 STEP 2:X(1)=Q(LP):Y(1)=R(LP):Z(1)=S(LP)
2060 X(5)=Q(LP+1):Y(5)=R(LP+1):Z(5)=S(LP+1):GOSUB 2500
2070 IF LN=0 THEN LA=LA+1:? "LINE OFF SCREEN WINDOW=";LA:GOTO 2090
2078 IF OVS THEN PLOT OVS*(X(2)+HZ)+HZ-OVS*HZ,(Y(2)+HZ)/2:DRAWTO OVS*(X(4)+HZ)+H
Z-OVS*HZ,(Y(4)+HZ)/2:GOTO 2090
2080 PLOT X(2)+HZ,(Y(2)+HZ)/2:DRAWTO X(4)+HZ,(Y(4)+HZ)/2
2090 POKE 53279,0:NEXT LP:GOTO 2010
2500 FOR A=1 TO 5 STEP 4:G=X(A)+X(3):S=Y(A)+Y(3):J=Z(A)+Z(3):X(A)=G*L1+S*L4+J*L7
2510 Y(A)=G*L2+S*L5+J*L8:Z(A)=G*L3+S*L6+J*L9:NEXT A
2600 FOR A=1 TO 5 STEP 4:D(A)=0:D(A+1)=0:D(A+2)=0:D(A+3)=0
2610 IF X(A)<-Z(A) THEN D(A)=1
2620 IF X(A)>Z(A) THEN D(A+1)=1
2622 IF Y(A)<-Z(A) THEN D(A+2)=1
2623 IF Y(A)>Z(A) THEN D(A+3)=1
2626 NEXT A
2632 FOR A=1 TO 4:IF D(A)=0 THEN 2640
2636 IF D(A)=D(A+4) THEN 2668
2640 NEXT A
2645 FOR A=1 TO 4:IF D(A)=1 THEN 2675
2648 NEXT A
2655 FOR A=5 TO 8:IF D(A)=1 THEN 2680
2658 NEXT A
2660 LN=1:GOTO 2800
2668 LN=0
2670 RETURN
2675 A=1:S=5:GOTO 2694
2680 A=5:S=1
2694 IF D(A)=1 THEN 2725
2696 IF D(A+1)=1 THEN 2715
2698 IF D(A+2)=1 THEN 2742
2700 IF D(A+3)=1 THEN 2756
2706 GOTO 2660
2715 J=(Z(A)-X(A))/(X(S)-X(A)-Z(S)+Z(A))
2720 X(A)=J*(Z(S)-Z(A))+Z(A):Y(A)=J*(Y(S)-Y(A))+Y(A):Z(A)=X(A):GOTO 2600
2725 J=(Z(A)+X(A))/(X(A)-X(S)-Z(S)+Z(A))
2730 X(A)=J*(Z(A)-Z(S))-Z(A):Y(A)=J*(Y(S)-Y(A))+Y(A):Z(A)=-X(A):GOTO 2600
2742 J=(Z(A)+Y(A))/(Y(A)-Y(S)-Z(S)+Z(A))
2745 X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(A)-Z(S))-Z(A):Z(A)=-Y(A):GOTO 2600
2756 J=(Z(A)-Y(A))/(Y(S)-Y(A)-Z(S)+Z(A))
2760 X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(S)-Z(A))+Z(A):Z(A)=Y(A):GOTO 2600
2800 IF Z(1)=0 THEN Z(1)=1.0E-03
2850 IF Z(5)=0 THEN Z(5)=1.0E-03
2860 X(2)=X(1)/Z(1)*HZ:Y(2)=Y(1)/Z(1)*HZ:X(4)=X(5)/Z(5)*HZ:Y(4)=Y(5)/Z(5)*HZ:RET
URN
11000 COLOR 1:IF HZ=159 THEN PLOT 65,0:DRAWTO 65,159:DRAWTO 253,159:DRAWTO 253,0
:DRAWTO 65,0:RETURN
11010 COLOR COL:IF COL=0 THEN COLOR 1
11020 IF HZ=79 THEN PLOT 32,0:DRAWTO 32,79:DRAWTO 127,79:DRAWTO 127,0:DRAWTO 32,
0:RETURN
11030 PLOT 16,0:DRAWTO 16,39:DRAWTO 64,39:DRAWTO 64,0:DRAWTO 16,0:RETURN
END
```

L-5

LISTING OF CORNERS.3D PROGRAM #5

```
1 REM CORNERS.3D Program version 5
2 REM COPYRIGHT 1980 SEBREE'S:T. HAYS:ALL RIGHTS RESERVED
10 GRAPHICS 0:LS=39:POKE 710,0
50 POSITION 5,4:? "Screen Resolution?":POSITION 11,5:? "X , Y":POSITION 5,6:?
   "1)  80 X  40"
60 POSITION 5,7:? "2) 160 X  80":POSITION 5,8:? "3) 320 X 160 HIRES"
70 TRAP 70:INPUT SR:IF SR=1 THEN HZ=39:GP=5:GOTO 100
80 IF SR=2 THEN HZ=79:GP=7:GOTO 100
90 HZ=159:GP=8
100 TRAP 100:POSITION 4,11:? "FIELD OF VIEW  .1 WIDE--2.0 SMALL":INPUT V
110 DIM Q(79),R(79),S(79),X(5),Y(5),Z(5),D(8),C$(3):FOR TR=1 TO 78:READ U:Q(TR)=
    U:NEXT TR
120 FOR PY=1 TO 78:READ I:R(PY)=I:NEXT PY:FOR PR=1 TO 78:READ E:S(PR)=E:NEXT PR
130 DATA -30,-30,-20,-20,-10,-10,0,0,10,10,20,20,30,30,30,-30,30,-30,30,-30,30,-
    30,30,-30,30,-30,30,-30
135 DATA -30,30
140 DATA 30,-30,30,-30,30,-30,30,-30,30,-30,30,30,20,20,10,10,0,0,-10,-10,-20,-2
    0,-30,-30,0,0,0,0,0,0
150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-10,-10,-20,-20
    ,-30,-30,-40,-40,-50,-50
170 DATA -60,-60,0,-60,0,-60,0,-60,0,-60,0,-60,0,-60,0,-60,-60,-60,-50,-50,-40,-
    40,-30,-30,-20,-20,-10
180 DATA -10,-60,0,-60,0,-60,0,-60,0,-60,0,-60,0
190 DATA 100,160,100,160,100,160,100,160,100,160,100,160,100,160,100,100,110,110
    ,120,120,130,130,140,140
200 DATA 150,150,160,160,160,160,160,160,160,160,160,160,160,160,160,160,160,160
    ,160,160,160,160,160,160
210 DATA 160,160,160,160,160,160,100,160,100,160,100,160,100,160,100,160,100,160
    ,100,100,110,110,120,120
220 DATA 130,130,140,140,150,150
400 OVS=0:TRAP 400:? "WANT OVERSCAN CORRECTION (OVS) YES?":INPUT C$:IF C$(1,1)="
    Y" THEN OVS=0.577
2000 GRAPHICS GP:SN=0.017453292:IF OVS THEN GOSUB 11000:IF GP=8 THEN POKE 710,0
2010 TRAP 2010:? CHR$(253):? "VIEWERS LOCATION X,Y,Z":INPUT U,I,E:X(3)=-U:Y(3)=I
    :Z(3)=-E:TRAP 2020
2020 ? "Heading, Pitch, & Bank (X,Y,Z)":INPUT H,P,B:P=-P:H=-H:? "COLOR":INPUT CO
    L:COLOR COL:IF ABS(B)>89 THEN 2020
2025 TRAP 2025:? "CLEAR Screen? YES?":INPUT C$:IF C$(1,1)="Y" THEN GRAPHICS GP:I
    F OVS THEN GOSUB 11000:IF GP=8 THEN POKE 710,0
2030 R6=COS(H*SN):R5=SIN(H*SN):R4=COS(B*SN):R3=SIN(B*SN):R2=COS(P*SN):R1=SIN(P*S
    N):L1=R4*R6+R5*R1*R3
2040 L2=-R6*R3+R5*R1*R4:L3=R5*R2*V:L4=R2*R3:L5=R2*R4:L6=-R1*V:L7=-R5*R4+R6*R1*R3
    :L8=R5*R3+R1*R4:L9=R2*R6*V
2050 FOR LP=1 TO LS*2-1 STEP 2:X(1)=Q(LP):Y(1)=R(LP):Z(1)=S(LP)
2060 X(5)=Q(LP+1):Y(5)=R(LP+1):Z(5)=S(LP+1):GOSUB 2500
```

L-6

```
2070 IF LN=0 THEN LA=LA+1:? "LINE OFF SCREEN WINDOW=";LA:GOTO 2090
2078 IF OVS THEN PLOT OVS*(X(2)+HZ)+HZ-OVS*HZ,(Y(2)+HZ)/2:DRAWTO OVS*(X(4)+HZ)+H
Z-OVS*HZ,(Y(4)+HZ)/2:GOTO 2090
2080 PLOT X(2)+HZ,(Y(2)+HZ)/2:DRAWTO X(4)+HZ,(Y(4)+HZ)/2
2090 POKE 53279,0:NEXT LP:GOTO 2010
2500 FOR A=1 TO 5 STEP 4:G=X(A)+X(3):S=Y(A)+Y(3):J=Z(A)+Z(3):X(A)=G*L1+S*L4+J*L7
2510 Y(A)=G*L2+S*L5+J*L8:Z(A)=G*L3+S*L6+J*L9:NEXT A
2600 FOR A=1 TO 5 STEP 4:D(A)=0:D(A+1)=0:D(A+2)=0:D(A+3)=0
2610 IF X(A)<-Z(A) THEN D(A)=1
2620 IF X(A)>Z(A) THEN D(A+1)=1
2622 IF Y(A)<-Z(A) THEN D(A+2)=1
2623 IF Y(A)>Z(A) THEN D(A+3)=1
2626 NEXT A
2632 FOR A=1 TO 4:IF D(A)=0 THEN 2640
2636 IF D(A)=D(A+4) THEN 2668
2640 NEXT A
2645 FOR A=1 TO 4:IF D(A)=1 THEN 2675
2648 NEXT A
2655 FOR A=5 TO 8:IF D(A)=1 THEN 2680
2658 NEXT A
2660 LN=1:GOTO 2800
2668 LN=0
2670 RETURN
2675 A=1:S=5:GOTO 2694
2680 A=5:S=1
2694 IF D(A)=1 THEN 2725
2696 IF D(A+1)=1 THEN 2715
2698 IF D(A+2)=1 THEN 2742
2700 IF D(A+3)=1 THEN 2756
2706 GOTO 2660
2715 J=(Z(A)-X(A))/(X(S)-X(A)-Z(S)+Z(A))
2720 X(A)=J*(Z(S)-Z(A))+Z(A):Y(A)=J*(Y(S)-Y(A))+Y(A):Z(A)=X(A):GOTO 2600
2725 J=(Z(A)+X(A))/(X(A)-X(S)-Z(S)+Z(A))
2730 X(A)=J*(Z(A)-Z(S))-Z(A):Y(A)=J*(Y(S)-Y(A))+Y(A):Z(A)=-X(A):GOTO 2600
2742 J=(Z(A)+Y(A))/(Y(A)-Y(S)-Z(S)+Z(A))
2745 X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(A)-Z(S))-Z(A):Z(A)=-Y(A):GOTO 2600
2756 J=(Z(A)-Y(A))/(Y(S)-Y(A)-Z(S)+Z(A))
2760 X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(S)-Z(A))+Z(A):Z(A)=Y(A):GOTO 2600
2800 IF Z(1)=0 THEN Z(1)=1.0E-03
2850 IF Z(5)=0 THEN Z(5)=1.0E-03
2860 X(2)=X(1)/Z(1)*HZ:Y(2)=Y(1)/Z(1)*HZ:X(4)=X(5)/Z(5)*HZ:Y(4)=Y(5)/Z(5)*HZ:RET
URN
11000 COLOR 1:IF HZ=159 THEN PLOT 65,0:DRAWTO 65,159:DRAWTO 253,159:DRAWTO 253,0
:DRAWTO 65,0:RETURN
11010 COLOR COL:IF COL=0 THEN COLOR 1
11020 IF HZ=79 THEN PLOT 32,0:DRAWTO 32,79:DRAWTO 127,79:DRAWTO 127,0:DRAWTO 32,
0:RETURN
11030 PLOT 16,0:DRAWTO 16,39:DRAWTO 64,39:DRAWTO 64,0:DRAWTO 16,0:RETURN
END
```

L-7

This version of AUTO.3D has been 'MASHED' , that is, compacted to use the least amount of memory as possible. You will notice some drastic changes from the style of the first 3 Three-D programs. Numeric constants-- numbers that are used more than twice in the program, have been assigned to a variable. This variable uses only 1 BYTE for the number '1234' rather than 4 BYTES. This mashing process saved about 3.4K of memory in this program!!!

LISTING OF PROGRAM #6- AUTO.3D Demo program

```
1 REM AUTO.3D PROGRAM  VERSION 23.6B :Copyright '80 & '81 SEBREE'S COMPUTING:TIM
   HAYS- Programmer: Update- 3/24/81
2 REM SEBREE'S COMPUTING: 456 Granite Ave.: Monrovia, CA. 91016: Phone num. (213
  )- 359-8092 : All Rights Reserved.
3 Q0=9000:Q1=1:Q2=0:Q4=4:Q5=5:Q6=3:Q7=8000:Q8=8:Q9=6:Q10=7:Q12=9:Q13=10:Q14=2:Q1
6=16:Q17=17:Q18=2010:Q20=0.017453292:Q21=11000
4 Q22=2020:Q23=1.0E-03:Q27=0.497:Q28=20:Q32=2030:Q36=2035:Q38=3000:Q39=39:Q41=40
:Q42=30:Q44=15:Q47=25:Q49=32:Q50=50:Q51=13000
5 Q55=55:Q61=60:Q64=64:Q65=65:Q70=70:Q78=78:Q79=79:Q80=80:Q87=87:Q90=2090:Q91=90
:Q100=100:Q106=106:Q127=127:Q148=148:Q159=159
6 Q200=200:Q228=228:Q253=253:Q272=54272:Q286=54286:Q600=2600:Q710=710:Q755=755:Q
774=53774:Q998=9998:Q999=9999
9 DIM Q(228),R(228),S(228),X(5),Y(5),Z(5),D(8),C$(4):M9=Q1:GOSUB Q0
10 REM INITIALIZE
12 GRAPHICS Q2:POKE Q710,Q2:LS=Q39:LA=Q2:? ">AUTO.3D Prog. vrsn. 18.E":? "NOTE:3
2K REQUIRED for >HI-RES<."
15 OVS=Q2:? "WANT OVERSCAN (OVS) CORRECTION? YES?":INPUT C$:IF C$(Q1,Q1)="Y" THE
N OVS=0.577
20 ? " Want to run DEMO MODE? YES?":INPUT C$:IF C$(Q1,Q1)="Y" THEN SR=Q6:HZ=Q159
:GP=Q8:T=Q1:K=Q7:GOTO Q106
50 POSITION Q5,Q9:? "Screen REsolution?":POSITION 11,Q10:? "X  ,  Y":POSITION Q5
,Q8:? "1)   80  x  40"
60 POSITION Q5,Q12:? "2)  160  x  80":POSITION Q5,Q13:? "3)  320  x  160  HIRES"
70 TRAP Q70:INPUT SR:IF SR=Q1 THEN HZ=Q39:GP=Q5:GOTO Q100
80 IF SR=Q14 THEN HZ=Q79:GP=Q10:GOTO Q100
90 HZ=Q159:GP=Q8:IF SR<>Q6 THEN GOTO Q50
100 TRAP Q100:POSITION Q4,13:? "FIELD OF VIEW (.1 WIDE -- 2.0 SMALL)":INPUT V
105 TRAP 105:POSITION Q4,Q16:? "1) AUTOMATIC LOOPING":POSITION Q4,Q17:? "2) USER
 INPUT":INPUT T:K=Q18:IF T=Q1 THEN K=Q7
106 ? "Visible processing (1), or non-visibleprocessing at 34% faster speed (2)"
:INPUT PX:IF PX>Q14 THEN GOTO Q106
107 IF T=Q1 THEN V=0.5:GOTO K-Q13
109 ? " GETTING DATA..."
110 LS=Q39:FOR TR=Q1 TO Q78:READ U:Q(TR)=U:NEXT TR
120 FOR PY=Q1 TO Q78:READ I:R(PY)=I:NEXT PY:FOR PR=Q1 TO Q78:READ E:S(PR)=E:NEXT
 PR
130 DATA -30,-30,-20,-20,-10,-10,0,0,10,10,20,20,30,30,30,-30,30,-30,30,-30,30,-
30,30,-30,30,-30,30,-30,-30,30
140 DATA 30,-30,30,-30,30,-30,30,-30,30,-30,30,30,20,20,10,10,0,0,-10,-10,-20,-2
0,-30,-30,-30,-30,-30
150 DATA -30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30,-30
,-30,-30,-30
160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-10,-10,-20,-20
,-30,-30,-40,-40,-50,-50
170 DATA -60,-60,0,-60,0,-60,0,-60,0,-60,0,-60,0,-60,0,-60,-60,-60,-50,-50,-40,-
40,-30,-30,-20,-20,-10
180 DATA -10,-60,0,-60,0,-60,0,-60,0,-60,0,-60,0
190 DATA 100,160,100,160,100,160,100,160,100,160,100,160,100,160,100,100,110,110
,120,120,130,130,140,140
200 DATA 150,150,160,160,160,160,160,160,160,160,160,160,160,160,160,160,160,160
,160,160,160,160,160,160
210 DATA 160,160,160,160,160,160,160,160,100,160,100,160,100,160,100,160,100,160,100,160
,100,100,110,110,120,120
220 DATA 130,130,140,140,150,150
```

```
230 IF T=Q1 THEN RETURN
2000 GRAPHICS GP:SN=Q20:POKE Q710,Q2:LA=Q2:IF OVS THEN GOSUB Q21:IF GP<>Q8 THEN
POKE Q710,Q148
2010 TRAP Q18:? CHR$(Q253):? "VIEWERS LOCATION X,Y,Z":INPUT U,I,E:X(Q6)=-U:Y(Q6)
=I:Z(Q6)=-E:TRAP Q22
2020 ? "Heading, Pitch, & Bank (X,Y,Z) ":INPUT H,P,B:P=-P:? "COLOR":H=-H:INPUT C
OL:COLOR COL:IF ABS(B)>89 THEN GOTO Q22
2025 TRAP 2025:? "CLEAR screen? YES?":INPUT C$:IF C$(Q1,Q1)="Y" THEN GRAPHICS GP
:IF OVS THEN GOSUB Q21
2027 POKE Q710,Q2:IF GP<>Q8 THEN POKE Q710,Q148
2030 IF PX=Q14 THEN NM=PEEK(Q286):IR=PEEK(774):DM=PEEK(Q272):POKE Q272,Q2:POKE
Q774,Q2:POKE Q286,Q2
2035 R6=COS(H*SN):R5=SIN(H*SN):R4=COS(B*SN):R3=SIN(B*SN):R2=COS(P*SN):R1=SIN(P*S
N):L1=R4*R6+R5*R1*R3
2040 L2=-R6*R3+R5*R1*R4:L3=R5*R2*V:L4=R2*R3:L5=R2*R4:L6=-R1*V:L7=-R5*R4+R6*R1*R3
:L8=R5*R3+R1*R4:L9=R2*R6*V
2050 FOR LP=Q1 TO LS*Q14-Q1 STEP Q14:X(Q1)=Q(LP):Y(Q1)=R(LP):Z(Q1)=S(LP):POKE 77
,Q2
2060 X(Q5)=Q(LP+Q1):Y(Q5)=R(LP+Q1):Z(Q5)=S(LP+Q1):GOSUB 2500
2070 IF LN=Q2 THEN LA=LA+Q1:? "LINE OFF SCREEN WINDOW = ";LA:GOTO Q90
2075 IF P1=10030 THEN POKE Q710,Q148:COLOR INT(Q6*RND(Q1)+Q1)
2078 IF OVS THEN PLOT (OVS*(X(Q14)+HZ)+HZ-OVS*HZ)*M9,(Y(Q14)+HZ)/Q14:DRAWTO (OVS
*(X(Q4)+HZ)+HZ-OVS*HZ)*M9,(Y(Q4)+HZ)/Q14:GOTO Q90
2080 PLOT (X(Q14)+HZ)*M9,(Y(Q14)+HZ)/Q14:DRAWTO (X(Q4)+HZ)*M9,(Y(Q4)+HZ)/Q14
2090 POKE 53279,Q2:RN=RN+Q1:? "LINE #";RN:NEXT LP:IF P1>10060 THEN GOTO P2
2095 IF K=Q18 AND PX=Q14 THEN POKE Q272,DM:POKE Q774,IR:POKE Q286,NM
2100 GOTO K
2500 FOR A=Q1 TO Q5 STEP Q4:G=X(A)+X(Q6):S=Y(A)+Y(Q6):J=Z(A)+Z(Q6):X(A)=G*L1+S*L
4+J*L7
2510 Y(A)=G*L2+S*L5+J*L8:Z(A)=G*L3+S*L6+J*L9:NEXT A
2600 FOR A=Q1 TO Q5 STEP Q4:D(A)=Q2:D(A+Q1)=Q2:D(A+Q14)=Q2:D(A+Q6)=Q2:IF X(A)<-Z
(A) THEN D(A)=Q1
2620 IF X(A)>Z(A) THEN D(A+Q1)=Q1
2622 IF Y(A)<-Z(A) THEN D(A+Q14)=Q1
2623 IF Y(A)>Z(A) THEN D(A+Q6)=Q1
2626 NEXT A:FOR A=Q1 TO Q4:IF D(A)=Q2 THEN 2640
2636 IF D(A)=D(A+Q4) THEN 2668
2640 NEXT A:FOR A=Q1 TO Q4:IF D(A)=Q1 THEN 2675
2648 NEXT A:FOR A=Q5 TO Q8:IF D(A)=Q1 THEN 2680
2658 NEXT A
2660 LN=Q1:GOTO 2800
2668 LN=Q2:RETURN
2675 A=Q1:S=Q5:GOTO 2694
2680 A=Q5:S=Q1
2694 IF D(A)=Q1 THEN 2725
2696 IF D(A+Q1)=Q1 THEN 2715
2698 IF D(A+Q14)=Q1 THEN 2742
2700 IF D(A+Q6)=Q1 THEN 2756
2706 GOTO 2660
2715 J=(Z(A)-X(A))/(X(S)-X(A)-Z(S)+Z(A)):X(A)=J*(Z(S)-Z(A))+Z(A):Y(A)=J*(Y(S)-Y(
A))+Y(A):Z(A)=X(A):GOTO Q600
2725 J=(Z(A)+X(A))/(X(A)-X(S)-Z(S)+Z(A)):X(A)=J*(Z(A)-Z(S))-Z(A):Y(A)=J*(Y(S)-Y(
A))+Y(A):Z(A)=-X(A):GOTO Q600
2742 J=(Z(A)+Y(A))/(Y(A)-Y(S)-Z(S)+Z(A)):X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(A)-Z(
S))-Z(A):Z(A)=-Y(A):GOTO Q600
2756 J=(Z(A)-Y(A))/(Y(S)-Y(A)-Z(S)+Z(A)):X(A)=J*(X(S)-X(A))+X(A):Y(A)=J*(Z(S)-Z(
A))+Z(A):Z(A)=Y(A):GOTO Q600
2800 IF Z(Q1)=Q2 THEN Z(Q1)=Q23
2850 IF Z(Q5)=Q2 THEN Z(Q5)=Q23
2860 X(Q14)=X(Q1)/Z(Q1)*HZ:Y(Q14)=Y(Q1)/Z(Q1)*HZ:X(Q4)=X(Q5)/Z(Q5)*HZ:Y(Q4)=Y(Q5
)/Z(Q5)*HZ:RETURN
```

```
3000 LS=114:? " GETTING DATA FOR SPACE SHUTTLE...":LA=0:RN=0
3020 FOR TR=Q1 TO Q228:READ U:Q(TR)=U:NEXT TR:FOR PY=Q1 TO Q228:READ I:R(PY)=I:N
EXT PY:FOR PR=Q1 TO Q228
3025 READ E:S(PR)=E:NEXT PR:DATA -2,2,2,4,-2,-4,4,15,-4,-15,15,24,-15,-24,24,42,
-24,-42,42,44,-42,-44,44,12,-44,-12,0,0
3050 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0
3060 DATA -5,-4,5,4,-4,4,-5,-2,5,2,2,-2,-2,-4,2,4,2,8,-2,-8,8,8,-8,-8,8,3,-8,-3,
-3,3,7,3,3,-3,-3,-7,-7,-6
3070 DATA 7,6,10,10,10,10,10,10,10,10,10,10,-10,-10,-10,-10,-10,-10,-10,-10,-10,
-10,8,8,-8,-8,8,8,-8,-8
3080 DATA 2,1,-2,-1,2,3,-2,-3,3,3,-3,-3,2,3,-2,-8,8,8,-8,-8,8,3,-8,-3,3,10,-9,-1
0,10,8,-10,-8,43,12,-43,-12
3090 DATA 0,0,0,4,4,0,0,-4,-4,0,4,2,-4,-2,0,0,8,6,-8,-6,4,4,-4,-4,1,2,-1,-2,4,4,
-4,-4,8,4,4,4,4,1,-8,-4
4000 DATA -4,-4,-4,1,8,8,-8,-8
4020 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-3,0,0,2,2,-1,-1,-
3,1,4,3,4,2,1,1,-1
4030 DATA -3,-6,-6,-9,-9,-15,-15,-19,-19,-21,-21,-21,-21,-24,-24,-48,-48,-49,-49
,-27,-27,-23,-21,-1,4,4
4040 DATA 4,2,-42,-42,-42,-29,-29,-29,-14,-15,-14,-15,-15,-15,-12,-19,-12,-19,-1
9,-19,-19,-15,-19,-15
4050 DATA -3,-7,-3,-7,-13,-13,-13,-13,-13,-18,-13,-18,-18,-18,-13,-18,-18,-18,-1
8,-13,-14,-17,-14,-17
4060 DATA -1,3,3,3,3,-1,3,6,5,7,-1,3,3,3,3,-1,3,6,5,7,0,-5,0,-5,-5,-1,-5,-1,-19,
-20,-19,-20,-19,-22,-19,-22
4070 DATA -22,-22,-22,-22,-19,-13,-19,-13,-13,-12,-13,-12,-13,-16,-13,-16,-16,-1
6,-16,-16,-15,-11,-15,-11
4080 DATA 0,0,0,0,-18,-21,-21,-17,-17,-13,-13,-17,-17,-21,-17,-16,-17,-16,-13,-1
3,-7,-6,-7,-6,-11,-10
4090 DATA -11,-10,-11,-10,-11,-10,-3,-2,-3,-2,-7,-11,-11,-2,-2,-11,-7,-11,-11,-2
,-2,-11,-12,0,-12,0
5010 DATA 100,100,100,104,100,104,104,157,104,157,157,170,157,170,170,189,170,18
9,189,202,189,202,202,205
5020 DATA 202,205,105,114,114,113,113,105,105,105,110,112,111,110,124,103,103,10
0,100,101,101,103,103,116
5030 DATA 116,119,119,122,122,193,193,194,194,224,224,242,242,226,226,208,208,21
0,217,194,194,124,242,228
5040 DATA 228,220,220,224,126,123,126,123,123,123,126,124,126,124,124,124,124,12
1,122,121,101,124,101,124
5050 DATA 132,189,132,189,132,132,132,132,132,132,189,191,191,191,191,189,124,13
2,124,132,168,169,169,179
5060 DATA 179,180,175,174,175,173,168,169,169,179,179,180,175,174,175,173,132,16
8,132,168,168,193,168,193
5070 DATA 193,218,193,218,193,196,193,196,196,218,196,218,193,193,193,193,193,21
8,193,218,193,196,193,196
5080 DATA 196,221,196,221,221,217,221,217,138,201,138,201,218,226,226,227,227,22
8,228,227,227,226,227,219
5090 DATA 227,219,228,220,229,221,229,221,228,220,228,220,228,220,228,220,221,23
0,221,230,229,228,228,230
6000 DATA 230,228,229,228,228,230,230,228,218,220,218,220
6010 X(Q6)=Q2:Y(Q6)=180:Z(Q6)=Q2:P=-Q61:B=31:H=-35:COLOR Q1:P2=6020:V=0.8:GRAPHI
CS Q8:POKE Q710,Q2:IF OVS THEN GOSUB Q21
6015 POKE Q87,Q10:M9=Q27:COLOR Q1:GOTO Q32
6020 POKE Q87,Q8:M9=Q1:X(Q6)=Q2:Y(Q6)=Q5:Z(Q6)=Q2:P=Q23:B=Q4:H=Q23:P2=6030:PLOT
Q2,Q80:DRAWTO 319,Q80:PLOT Q159,Q2:DRAWTO Q159,Q159:GOTO Q36
6030 X(Q6)=-Q100:Y(Q6)=Q5:Z(Q6)=Q2:P=Q28:B=Q2:H=-57:P2=6040:GOTO Q36
6040 POKE Q87,Q10:M9=Q27:COLOR Q14:X(Q6)=-Q91:Y(Q6)=Q70:Z(Q6)=Q2:P=-Q41:B=Q2:H=-
Q13:P2=6045:GOTO Q36
6045 M9=Q1:IF PX=Q14 THEN ? CHR$(Q253):POKE Q272,DM:POKE Q774,IR:POKE Q236,NM:GR
APHICS Q41:POKE Q710,Q2:? CHR$(Q253)
```

```
6050 ? CHR$(Q253):? CHR$(Q253):GOSUB Q999:GOSUB Q999
6060 GRAPHICS Q2:POKE Q710,Q2:? :? :? "MOVEBACKANDFORTHBETWEENGRAPHICSANDTEXT":F
OR O=Q1 TO 500:NEXT O:FOR O1=Q2 TO Q6
6065 POSITION Q1,23:? "Use more colors in HI-RES !!!"
6070 GRAPHICS Q41:POKE Q710,Q2:FOR O=Q2 TO 170:NEXT O:GRAPHICS Q2:POKE Q710,Q2:?
 :? :?
6080 ? "SAVE YOUR TEXT WINDOWS !!! !!!":FOR O=Q2 TO 150:NEXT O:NEXT O1
6090 RESTORE :GRAPHICS Q2:POKE Q710,Q2:? :? :? "THE AUTO PROGRAM LOOP IS COMPLET
ED AS OF NOW, IT SHALL CONTINUE TO RUN."
6095 ? CHR$(Q253):FOR O=Q2 TO 450:NEXT O
7000 GOSUB Q0
7990 GRAPHICS GP:POKE Q710,Q2:SN=Q20:P1=10000:GOSUB 110:V=0.4:P2=Q38
7995 DM=PEEK(Q272):IR=PEEK(Q774):NM=PEEK(Q286)
8000 ? CHR$(Q253):? "VIEWERS LOCATION X,Y,Z":GOSUB Q998:GOSUB P1:? "?";X(Q6);","
;Y(Q6);",";Z(Q6)
8005 IF PX=Q14 THEN POKE Q272,DM:POKE Q774,IR:POKE Q286,NM:? CHR$(Q253):GRAPHICS
 Q41:POKE Q710,Q2:? CHR$(Q253)
8007 IF B=-Q42 THEN GRAPHICS Q39:GOSUB 12000
8010 ? "HEADING, PITCH, & BANK":GOSUB Q998:? "?";H;",";P;",";B:RN=Q2
8020 ? "COLOR":GOSUB Q998:? "?1":? "CLEAR screen? YES?":GOSUB Q998:? "YES":P1=P1
+Q13:IF P1>10070 THEN GOTO Q38
8040 GOSUB Q999:GRAPHICS GP:IF OVS THEN GOSUB Q21
8050 IF GP=Q8 THEN POKE Q710,Q2
8060 GOTO Q32
9000 ? CHR$(Q253):GRAPHICS Q2:O=Q3+PEEK(742)*256+PEEK(741):SETCOLOR Q14,Q14,04:S
ETCOLOR Q1,Q14,Q13
9010 SETCOLOR Q2,Q14,Q44:SETCOLOR Q4,Q9,Q16:POKE Q710,113:POSITION Q1,Q14:? "SEB
REE'S COMPUTING":POSITION 26,Q5:? "PRESENTS"
9020 POSITION Q4,Q10:? "3-DIMENSIONAL     COMPUTER GRAPHICS":POSITION Q6,Q17:? "
(C)'80 Sebree's Computing-ATARI 800"
9030 POKE O+Q14,Q10:POKE O+Q3,Q3:POKE O+Q3,Q3:POKE O+Q12,Q10:POSITION Q14,Q13
9040 ? CHR$(Q253):FOR O=Q1 TO Q5:FOR A1=Q1 TO Q55:NEXT A1:POKE Q755,255:FOR A=Q1
 TO Q61:NEXT A:POKE Q755,Q2:NEXT O
9050 SETCOLOR Q14,Q4,Q44:SETCOLOR Q4,Q4,Q44:SETCOLOR Q2,Q14,Q16:FOR A=Q1 TO Q200
:NEXT A:RETURN
9998 FOR A=Q1 TO 120:NEXT A:RETURN
9999 ? "PLOT FINISHED, & CONTINUING":FOR A=Q1 TO Q200:NEXT A:RETURN
10000 X(Q6)=-Q31:Y(Q6)=Q70:Z(Q6)=Q2:P=-Q28:B=Q55:H=-Q42:COLOR Q1:RETURN
10010 X(Q6)=Q2:Y(Q6)=Q42:Z(Q6)=Q2:P=Q2:B=Q61:H=Q2:RETURN
10020 X(Q6)=-95:Y(Q6)=Q70:Z(Q6)=Q2:P=-Q13:B=-Q55:H=-Q42:GP=Q10:HZ=Q79:RETURN
10030 X(Q6)=Q2:Y(Q6)=Q42:Z(Q6)=Q2:P=Q2:B=-Q42:H=Q2:GP=Q8:HZ=Q159:COLOR Q1:RETURN

10040 X(Q6)=-Q61:Y(Q6)=-Q50:Z(Q6)=Q2:P=Q47:B=Q61:H=-45:RETURN
10050 X(Q6)=Q2:Y(Q6)=Q42:Z(Q6)=Q2:P=Q2:B=Q2:H=Q2:RETURN
10060 X(Q6)=-Q31:Y(Q6)=Q70:Z(Q6)=Q2:P=-Q28:B=-Q55:H=-Q47:RETURN
11000 COLOR Q1:IF HZ=Q159 THEN PLOT Q65,Q2:DRAWTO Q65,Q159:DRAWTO Q253,Q159:DRAW
TO Q253,Q2:DRAWTO Q65,Q2:RETURN
11010 COLOR COL:IF COL=Q2 THEN COLOR Q1
11020 IF HZ=Q79 THEN PLOT Q49,Q2:DRAWTO Q49,Q79:DRAWTO Q127,Q79:DRAWTO Q127,Q2:D
RAWTO Q49,Q2:RETURN
11030 PLOT Q16,Q2:DRAWTO Q16,Q39:DRAWTO Q64,Q39:DRAWTO Q64,Q2:DRAWTO Q16,Q2:RETU
RN
12000 ? "ROTATE the COLORS of your scene in anycombination you want!!!"
12010 FOR TR=Q2 TO Q42:SETCOLOR Q14,Q8,Q14:SETCOLOR Q1,Q8,Q5:SETCOLOR Q2,Q8,Q8:G
OSUB Q51:SETCOLOR Q2,Q8,Q14:SETCOLOR Q14,Q8,Q5
12020 SETCOLOR Q1,Q8,Q8:GOSUB Q51:SETCOLOR Q1,Q8,Q14:SETCOLOR Q2,Q8,Q5:SETCOLOR
Q14,Q8,Q8:GOSUB Q51:NEXT TR
12030 POKE 708,Q41:POKE 709,202:POKE Q710,Q148:POKE 711,Q70:POKE 712,Q2:RETURN
13000 FOR PY=Q2 TO 19:NEXT PY:RETURN
```

>>>*GRAPHICS EDITOR!!*<<<   ATARI 800

16K min. 40K Preferred. NOW, both 2-D and 3-D scenes can be designed with a JOYSTICK, and then saved to disk!! These scenes can then be loaded in later to be edited before you save it again under another name!! All of this can be done using ANY graphics mode! But that's not all!!! You can save entire screens OR just individual images in 2-D or 3-D!! You have the option of giving the new scenes different file names, a MENU of disk file names is shown on the screen as you choose the file name. A HELP option is included should you have any trouble with ANY operation. If you decide to use the 3-D option, you may change the 3-D view(s) of the object(s) on the screen. Uses Player-Missile-Graphics for the fast acting 'Cursor'. POWERFUL, and WELL DOCUMENTED!! DISK version is recommended. (--add $5.00--). With 5 programs. Only $29.95 + $1.50 p&h on cassette.

>>TRS-80 COLOR COMPUTER EXTENDED BASIC >>3-D C.C. GRAPHICS PACKAGE!!<< NOW, you can get our popular 3-D Graphics Package for your 16K COLOR COMPUTER!!! Design you own graphics with a joystick & view these images from any angle you want!! software selectable Screen Resolutions, Colors, Viewing angles, Rotation, Object erase & replacement, Wide-angle or Telephoto views, along with ALL of the required 3-D operations to change the viewers location. WITH MANUAL AND LISTINGS!! ONLY $24.95 + $1.50 p&h. on cassette.

SEBREE'S COMPUTING PRESENTS
SOFTWARE FOR THE ATARI* 400/800 PERSONAL COMPUTER.

Atari 800
Software

SEBREE'S COMPUTING:
456 Granite Ave.
Monrovia, Ca 91016
Phone- (213)-359-8092

*Atari 800**
Software

## WHEN ORDERING, BE SURE TO INDICATE- PROGRAMS FOR 'ATARI' COMPUTER

All programs are supplied on tape unless otherwise specified. A operation manual is supplied with each program along with FULL LISTINGS of the programs!!! The photographs below were taken from the actual programs to give you a visual idea of what the program is like, however, the colors had to be changed for photographic purposes, the colors may be entirely reversed. The memory requirements have been listed in front of each program, this is the absolute minimum, some programs (DOWN THE TRENCH, for example) have many versions included on the tape, so a particular version of a program you want to use MAY require more memory than specified. NOTE: Updated programs are made from time to time. We will send you an updated tape at a low cost of $3.00 , (for postage and duplication costs) when you RETURN the ORIGINAL program tape.

TO HAVE YOUR PROGRAM(S) PUT ON A DISKETTE, ENCLOSE $5.00 EXTRA.

16K WUMPUS ADVENTURE: This is a VERY dynamic version of the popular game 'WUMPUS'. There is the addition of graphics and sound effects!!! The program takes up all of the 16K memory. The playing field is very alive with deadly CAVE SNAKES, SUPER BATS which are hard to see in the dim light of the caverns, and other assorted hazards you would expect to find in a caverns. The object of the game is to search through a maze of caverns for the mythical 'WUMPUS'. The WUMPUS is very hard to find and , of course, can move himself and possibly attack YOU!! There are many things to be careful of in a caverns, hazards abound everywhere. To catch the WUMPUS, you must shoot him with one of your 3 arrows, or throw a rock at him to knock him out, but throwing rocks are not very accurate (if you run out of arrows). Beware of the WUMPUS, he may decide to have you for lunch!!! only- $14.95 +.95 p&h.
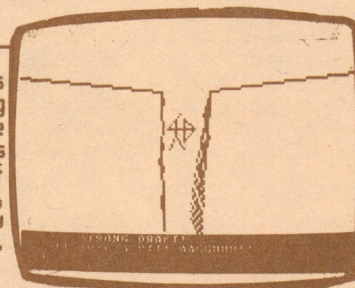

24K WUMPUS ADVENTURE- EXPANDED VERSION! This game is a much expanded version of the 16K WUMPUS ADVENTURE above!! The graphics are all done in HI-RESolution in multi-colors!!! Enhanced sound effects, along with an extra 8K more of hazards and suprises!! This game is much more like an ADVENTURE than a WUMPUS!! There are graphic CAVE-INs, KILLER DWARVES, SNEAKY CAVE TROLLS (they steal your arrows and supplies), options on entering certain cave numbers at the start of the game, and a snake bite kit available. The graphic illustrations are much better than before. The cavern set-up is always different at the start of the game,but you have the option of playing the same setup as before if you somehow killed early in the game. There are possibilities of finding an arrow left by a previously eaten explorer, and finding throwing rocks when you are out of arrows. When you finally do get the WUMPUS, the musical theme from the olympics is played just for you! Only- $17.95 +.95 p&h.
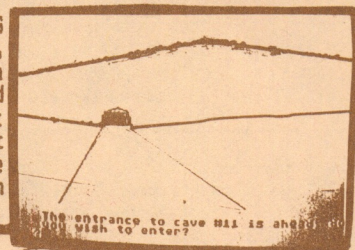
8K & 16K & 24K DOWN THE TRENCH: This is one of our best selling- most popular game programs!!! It is a set of 2 programs that include 8K, 16K, and 24K versions!! The second program has the options in it for 16K or 24K operation (HI-RES in multi-colors!!). The theme of these programs is to keep yourself from getting hit from a laser post while flying your space-craft down the TRENCH of the dreaded DEATH STAR!! The 24K version includes DARTH VADER on your tail (shown on the AFT VIEW) trying to shoot you down, DO NOT STAY STILL!! You have to fly through this mess of LASER fire until you near your target, the exhaust port. This is where your on - board attack computer engages itself and you have to line-up your target and fire at the right moment, pull your ship up and out of the TRENCH and away from the DEATH STAR, and watch it destroy itself (if your shot was accurate!!) NOTE: This program set is definetly NOT the same as the one published in SOFTSIDE (tm), as this set is VERY much expanded and enhanced!! Requires 1 Joystick. Only- $16.95 + .95 p&h.
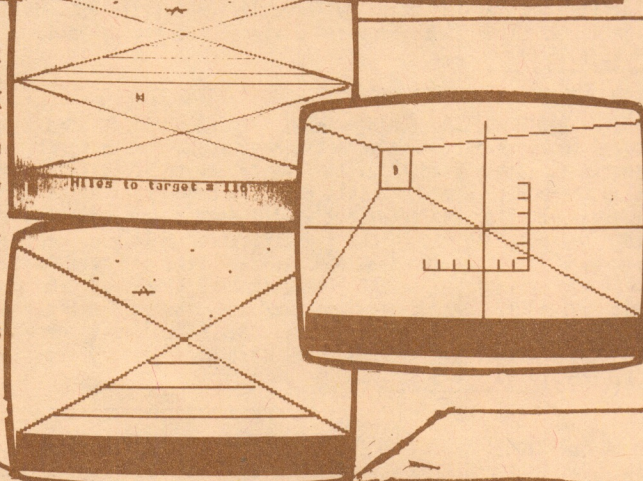
8K BATTLING TANKS: This is a two player TANK battle game that uses graphics in a unique way so that the TANKS go different speeds over different terrain. If your tank is fired upon, you still have time to move while the shot is advancing. If your shot runs into an obstacle, it harmlessly explodes! The rotation matrix is extremely fast! The graphics are fast enough to swear it is almost real- time!! But, remember, TANKS don't move very fast anyway. You have the option of setting the maximum score at the start of the game, and the computer shows the score at regular intervals, and shows who is the winner at the finish. This another one of our VERY popular graphics games!! The game requires TWO joysticks and TWO human players. Only- $9.95 +.95 p&h.
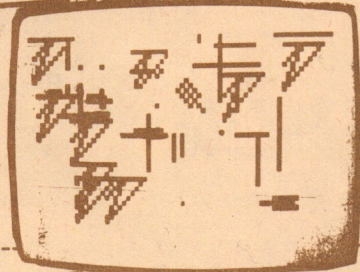

SOFTWARE FOR
THE- 'ATARI 800'

Atari 800
Software

Software

SEBREE'S COMPUTING
SOFTWARE FOR THE ATARI- Continued

---

_16K 3-D RED BARON DOGFIGHT/ FLIGHT SIMULATOR!! The title practically speaks for itself! This is one of our best selling games that requires that you learn to fly before you DOGFIGHT! The graphics are supurb, all done in the HI-RESolution mode in multi- colors!!! The sound effects are very realisitc, and the overall capabilities are very fun to learn. Entirely programed in BASIC but executes quite fast. The program will only re-fresh (re-draw) that part of the screen that requires it, so screen refreshing changes from SIX cycles per second to 2.25 cycles per second. This is an updated version (updated on 3/26/81) expanded. You are flying behind the RED BARON, but catching up to him. If you don't line him up and shoot him down soon, his tail-gunner will start shooting at you! But remember, his craft doesn't stay stationary!A RADAR, ALTIMETER & BANKING METER, and number of wins continually shown at the bottom. Requires 1 joystick, and one human pilot. Only- $16.95 +.95 p&h.
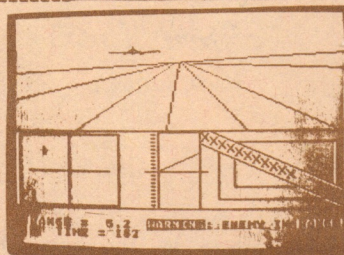
_8K & 16K PROGRAM SAMPLER #1: This set of programs is a nice way to see the quality of our programs at a low price.There are FOUR (4) programs in this package!! Programs are: EXOTIC CIRCLE PLOTTER, generates a wide variety of outputs in a variety of types and colors of circles. CRASH COURSE, try to keep your speeding car within the boundaries of the road by using your joystick and get the highest mileage, then try pressing your trigger and going into OVERDRIVE!! HI-RES SCRIBBLER, uses one player with a joystick to allow you to draw anything you want on the screen —from 1 to 4 (FOUR!) COLORS! VIDEO MOIRE, a graphic program where the ATARI will display a MOIRE pattern in a variety of different modes, all HI-RES in 4 colors!! Also includes a RANDOM VECTOR GENERATOR within the same program. This GENERATOR puts up some interesting designs using non-standard line drawing techniques. A good value! Requires 1 joystick for two of the programs. >>> ONLY!-- $9.95 +.95 p&h.


8K SUBMARINE MINEFIELD navigator: This game can be played by one OR two players using joysticks! You will have to move through a VERY dense minefield with moving depthcharges traveling all around you! Then, to make matters worse, a RADAR- SEARCH team appears in a boat above you. If their radar finds you, they will destroy your ship with a laser!! The object of the game is to navigate your submarine through the minefield and other obsticles to get to the right side of the playfield first, in the least amount of time. Your submarines have full up- down- foreward- reverse maneuverability. The computer will keep score from one game to the next and total the scores at the end of each round, showing the winner when you have completed the number of rounds you specified at the beginning of the game. Requires two joysticks, and two human navigators. Only- $9.95 +.95 p&h.

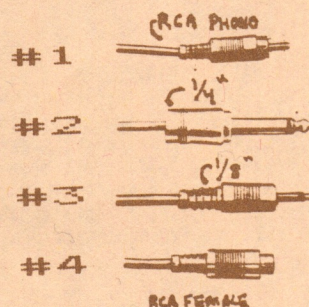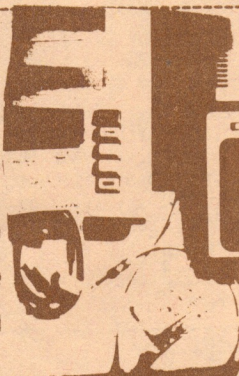---
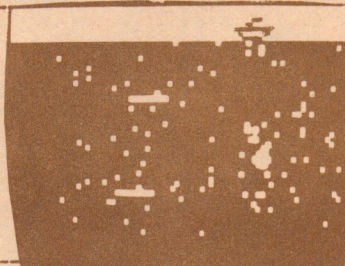
8K & 16K BALLOON POP & PYLON RACER: This is a set of programs in which you are flying a radio- controlled plane. In BALLOON POP, you try to run into and pop as many balloons as you can. As your plane goes off the screen on the right side, (where the flying balloons are) it appears again on the left. In PYLON RACER, you have to complete the 10 laps around the track, without running into any other planes, the ground, a pylon, or a flag- holder!! These programs are tailored after the actual AMA (American Modelers Association) competition rules!! BALLOON POP requires 8K RAM and 1 joystick. PYLON RACER requires 16K RAM and 1 joystick. PYLON RACER includes 2 views of the playfield, to simulate 3-D and give you better control around the pylons. Only- $9.95 +.95 p&h.

16K BASIC EDITOR / MINI- WORD PROCESSOR: This is a program designed to use with any printer hooked to the ATARI. The advertisement you are reading right now, was done using this BASIC EDITOR. The program is mainly used to allow anyone to enter text information into the computer and make changes using the built-in editor. This material can then be saved on cassette or DISK (designed for DISK) to be accessed later and print out in the format you will specify. The printing formats include margin width changes, and right- end justification. If you decide to specify right- end justification, the program will locate the last WORD in each line and then move it to the right, spacing itself equally until the line length (width) is equivalent to the margin width you specified earlier. For instance, this text is 106 characters long per line, the margin width equals 106. Programmed entirely in BASIC, but the execution is extremely fast, the LONGEST wait there is (with exception of DISK operations) is less than — two (2) seconds. Requires more memory for large amounts of data. Requires 1 printer. >>ONLY-- $15.95 + .95 p&h.
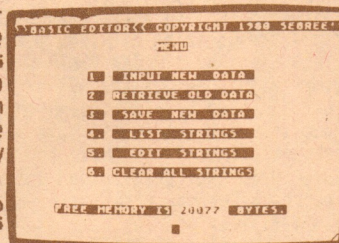
8K HARDWARE: DIRECT SOUND OUTPUT CABLE with software. You can now run the high quality sound from your ATARI sound synthesizer directly into your amplifier or home stereo system!! Just imagine playing STAR RAIDERS (tm) with true HI-FI sound!! Not the little sound that comes out of your TV speaker. You can then record this sound or further process it through your equipment. The cable plugs directly into the MONITOR jack on the side of your ATARI, therefore, it is restricted to the ATARI 800 ONLY. If you plan to use the ATARI SOUND SYNTHESIZER for true music, then you should not be without one of these cables. The demonstration software is a program that uses a joystick to control all of the sound variables graphically on the screen. This allows you to design some new sounds, and instantly be able to see what value each of the sound control variables is set at. When ordering, please specify which plug you need at the end of your cable, shown at right. Plug #1 is used for home stereo, #2 is used for an amplifier, #3 for cassette recorder, and #4 for mono-to-stereo conversion. Requires ATARI 800. ONLY-- $17.95 +.95 p&h.

#1   RCA PHONO
#2   C 1/4"
#3   C 1/8"
#4   RCA FEMALE

----

* Indicates Trademark(s) Atari , Inc.

>SOFTWARE for the ATARI$
PERSONAL COMPUTER SYSTEM.
Continued....

SEBREE'S COMPUTING
PRESENTS:
3-DIMENSIONAL COMPUTER GRAPHICS!!

FROM: SEBREE'S COMPUTING, 456 Granite Ave., Monrovia, Ca. 91016
Phone- (213)-359-8092

SEBREE'S COMPUTING is proud to announce a 3-Dimensional computer graphics package which will run on any ATARI PERSONAL COMPUTER SYSTEM under BASIC. There are TWELVE (12!!) programs included in this package!! ALL LISTINGS ARE GIVEN!!! This set of programs will run on an 8K machine up to a 48K machine, (last program requires 32K to run in HIRES). The base program allows you to view ANY object you so desire, from ANY location in space, even INSIDE of it!!! The software has been engineered so any person can quickly learn how to put in his own data and display it in the order that he wishes. The processing speed is very fast compared to other HIGH PRECISION 3-D algorithms. A data base with 237 XYZ points will execute in 21.5 seconds!!
The software is intended as a tutorial on the handling of exotic graphics through ATARI BASIC, therefore, anybody can use the 32 (THIRTY TWO!!) page manual to modify the software where it is shown, or leave it be so he can just control the view(s) of the objects he selects. Engineers can find the program package extremely useful in the design of solid objects. The art forms alone are extremely entertaining. The package goes through a whole chapter on controlling MULTI-COLOR graphics in HI-REsolution mode!!! Another section completely explains how to blank out the interrupts temporarily to obtain a 34% FASTER EXECUTION time!! A chapter of input examples is included, along with all the required information so that the user can do some VERY POWERFUL graphics programming. Software selectable screen resolution, multiple-color control, object erase and replacement, field of view control (telephoto or wide- angle), along with ALL of the required 3-D operations- HEADING, PITCH, & BANK (rotation), and changes of the VIEWERS LOCATION X, Y, & Z.
This package represents 6 months work under the ATARI BASIC making the floating point high precision 3D to 2D transformation generator as fast as possible, while condensed to use as little memory as possible: does anybody else sell an 8K 3-D program? Line clipping and pushing are performed: A line which intersects the screen, but the two points lie off the screen, will be 'pushed' and 'clipped' until it lies fully on the screen. Six of the 12 programs are on cassette (fills a C-30 cassette), listings and instructions are given for all the rest. All TWELVE programs plus DOS 2.0S and four (4) DISK handling programs, totaling 17 programs on the Diskette version. This set of programs has thousands of applications!!! It may very will be the most powerful, and most often used piece of software you own!! COMPLETELY DOCUMENTED!! PRICE FOR ALL OF THIS IS ONLY: $29.95 + $1.50 p&h (Supplied on cassette). Diskette version- add $5.00 for DISK and extra costs.
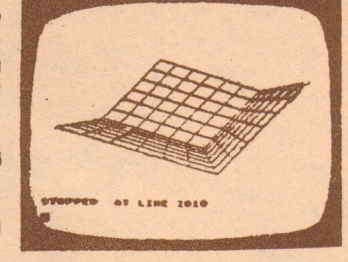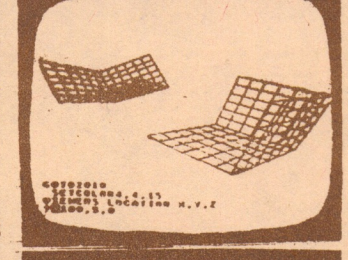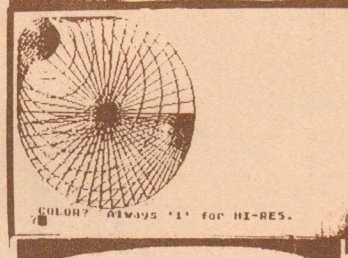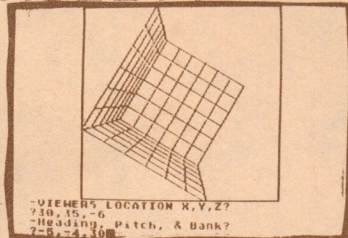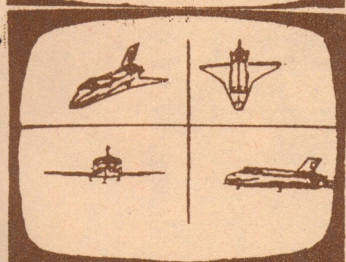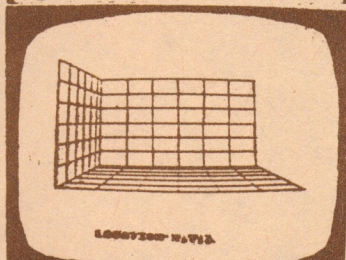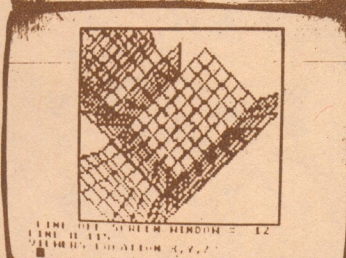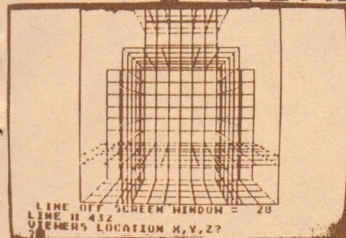
16K WORD GAMES: This program includes two word games called 'GUESSIT', and 'JUMBLE'. One or Two players. In GUESSIT, you try to guess what a word the computer has chosen is. The strategy is to 'narrow' down and eliminate as many letters as you can to come as close as you can to guessing the word. Even if you don't know the word, you will eventually get it through process of elimination. But it isn't that easy, you have the option of a time limit or number of guesses limit. In JUMBLE, you must un-scramble 4 words in order to get the main answer- word #5. A hint is given on the screen as to the answer word which is made up partly of the first 4 words. An asterisk (see photo) is in the places that correspond to the letters that make-up the answer word. If you can't get the answer, a GIVE-UP option is allowed. Instructions are also included on changing the words used to guess upon within the program. Lots of fun!! Only- $9.95 + .75 p&h.
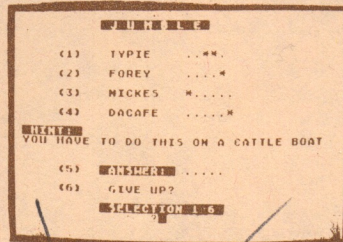
>>>>> HARDWARE: MEMORY EXPANSION FOR THE
ATARI 800 PERSONAL COMPUTER SYSTEM <<<<<

16K MEMORY CARD FOR ONLY $129.95 !
JUST LOOK AT ALL THESE 'EXTRAS'!!

QUALITY HARDWARE™

-------GOLD PLATING to inhibit corrosion!!!
------ELIMINATES costly oxidation failures!
----FASTER ACCESS TIME, 200 NS MEMORY (FASTER MEMORY)!!!
-----------------LOWEST COST PER BIT!
-------FULLY COMPATIBLE with current HARDWARE!
--------------FULL ONE YEAR WARRANTY!!
------EASY TO INSTALL, front and back are marked!
-----------BACKED BY 'QUALITY HARDWARE'
----GOLD MIND- All you have to remember is to plug it in!
>>>>>>>>>>> Just $129.95 + $1.25 postage from SEBREE'S.

Sebree's Computing
456 GRANITE AVE
MONROVIA CA 91016
(213) 359 8092

* Indicates Trademark(s) Atari , Inc.

TRS-80 POCKET COMPUTER SOFTWARE!!! :

2K POCKET COMPUTER- WUMPUS ADVENTURE!!! Play this popular game on your POCKET COMPUTER! The WUMPUS is the mythical creature that you seek. To find him, you must visit his home territory- a maze of caverns. There are many things that can go wrong in this caverns, though: hazards abound everywhere. You are trying to locate the WUMPUS when you suddenly fall into a PIT or get snatched by a SUPER BAT colony- placing you in a random room. There are many things that can happen in this caverns. You have 3 arrows in which to shoot your prey. The arrows can travel through from 1 to 5 rooms (you specify). The cave setup is randomly different every time, relying on the built —in RANDOM NUMBER GENERATOR (remember, your POCKET COMPUTER doesn't have one)!! But if you are killed early in the game, you have the option of playing the same setup as before. The cave setup STAYS the same during the period of the   current game (not just randomly changing). This way, you can predict which room the WUMPUS is in and act upon it. BEWARE the WUMPUS, he may decide to have you for lunch! Supplied on cassette with playing manual & listing. All this for only>>-- $7.95 + .95 p&h.

ATARI 800 DISK INVENTORY SYSTEM!!

16K ATARI 800 DISK INVENTORY SYSTEM! This system on DISK allows you to deep an up-to-date inventory of your   entire DISK library as it grows. The DOS used in the system is the new ATARI: DOS 2.0S system, included on the   diskette. You can now check your entire inventory, search for a program to see if it is already saved, and if you have a backup copy: warning you if you don't. You can print a listing of your entire DISK library and know which DISK a   particular program(s) is on. This is an extremely POWERFUL SET of programs to handle your DISK library. The program   set is : 'DISKDIR', disk directory program. 'DISKINV', disk inventory program. 'DISKMENU', a good MENU program for   your DISK operations. And, 'DISK01.DAT', a data file where your information is stored. 'DISKMENU', a good MENU program for   your DISK operations. And, 'DISK01.DAT', a data file where your information is stored. Of course, the two files for DOS 2.0S included, and an   'AUTORUN' file that will automatically boot up the menu program as soon as the DOS 2.0S is   loaded into memory! A MUST for anyone who has a large diskette library. ONLY- $19.95 +.95 p&h +$5.00 for the diskette.

TRS-80 COLOR COMPUTER SYSTEM SOFTWARE!!! AVAILABLE MAY 1981

16K COLOR COMPUTER-EXTENDED BASIC. WUMPUS ADVENTURE!! Now you can play this exciting game on your COLOR COMPUTER! This is much like the ATARI 24K version (Page ONE) with graphics and sound effects!! Read Page 1 for general info.   ONLY- $15.95 + .95 p&h.

16K COLOR COMPUTER-EXTENDED BASIC. 3-D RED BARON DOGFIGHT/ FLIGHT SIMULATOR!!! You can now play an exciting   3-D flight simulation game on your COLOR COMPUTER. See ATARI version Page 2 for general info. Requires 1 joystick, and one human pilot. ONLY- $16.95 + .95 p&h.

16K COLOR COMPUTER-EXTENDED BASIC. DOWN THE TRENCH!! Our most popular game program comes to the COLOR COMPUTER! See ATARI version Page 1 for more information. Requires 1 joystick and one courageous pilot. Only- $16.95 + .95 p&h.

16K COLOR COMPUTER-EXTENDED BASIC. BATTLING TANKS!! This is a 2 player version of the popular TANK war game. Similar to the ATARI version of this program, see the ATARI version, bottom of page # 1 for more information. Requires two joysticks and two human operators. Only- $9.95 + .95 p&h.

16K COLOR COMPUTER-EXTENDED BASIC. SUBMARINE MINEFIELD navigator!! Play this popular original game on your   TRS-80 COLOR COMPUTER. Very similar to the ATARI version. See ATARI version Page # 2 for more information. One OR Two players: Requires Joysticks. Only- $9.95 + .95 p&h.

NEW Programs for BOTH ATARI & TRS-80 COLOR COMPUTERS!
Available MAY 1981,
NOTE: When ordering, specify WHICH computer you want program(s) for.

24K ATARI or 16K COLOR COMPUTER-EXTENDED BASIC. TRIP TO JUPITER SPACE ADVENTURE!! This is a new space adventure. program that goes as follows: You have to successfully launch your spacecraft from EARTH and get on a correct   trajectory to JUPITER. But there are obsticles to overcome: An asteroid field shows up!! If you maneuver through the asteroid field successfully, you still have to make sure that you are on course. When you finally make it around the orbit of JUPITER, you have to be careful not to run into, or be thrown off course by on of the 13 moons!! SPECTACULAR graphics now show up as you try to bring your ship down to the planet to collect samples of the atmosphere, ground (if there is any) and the oceans. But now, you have to re-launch your craft (in high gravity) and bring it back in orbit to re-connect with your mothercraft, being careful not to destroy either ship by docking too hard!! Spectacular game!!! All this and a large playing manual for ONLY>>> ~~$~~ + .95 p&h. $19.95

16K ATARI or COLOR COMPUTER-EXTENDED BASIC. AIR-SEA WATCH COMMANDER!! You are stationed on a missile base off the coast of Pearl Harbor. A war has broken out and you must defend the nation's best fighting equipment by using   your missle-torpedoe launchers aimed at the enemy. The enemy may be in the sky or in the sea. Be sure to watch for those sneaky periscopes, otherwise you may get hit yourself from that subarine under it!!! The planes from up above can drop bombs, so be careful to use your ground to air interceptor missiles correctly. This game is much like an expanded arcade SEA-WOLF with AIR-COMMAND- like properties. Fast graphics and lots of fun!! Requires one joystick and one human commander. All for only- $16.95 + .95 p&h.

16K ATARI or COLOR COMPUTER-EXTENDED BASIC. FLIGHT SIMULATOR!! This is a GRAPHIC flight simulator for one player! You have to successfully take- off and get to a specified altitude. If you fly too low or too high, collision may   result with another aircraft!! You are to navigate a given number of miles to the next airport. Mileage varies depending on the level of difficulty you chose. During this flight, you have to be careful of the mountains coming- up in the distance. You must fly OVER these mountains thriftly, you don't want to burn up all your fuel!! After you have gone over the mountains, you must re- initialize your destination and be careful not to overshoot it. You will then see your destination airport, call in to the conning tower and ask for clearance. If you still have enough fuel (snicker) you shall attempt to land safely. If you are out of fuel, you have to try to land 'DEAD STICK'!! Great graphics!! Requires 1 joystick and one good pilot. All for only- $17.95 + .95 p&h.